

**O'ZBEKISTON OLIY VA O'RTA MAXSUS TA'LIM VAZIRLIGI**

**SAMARQAND DAVLAT UNIVERSITETI**

**Mexanika-matematika fakulteti**  
**«Axborotlashtirish texnologiyalari» kafedrası**

5140800 – **«Amaliy matematika va informatika»**  
yo'nalishining 2-kurs talabalari uchun

**Axadov Akmal Rustamovich**

## **ALGORITMLAR NAZARIYASI**

fani bo'yicha

# **Ma'ruzalar matni**

SAMARQAND -2011

### **SUZ BOSHI**

Hozirgi kunda biror bir sohada ishni boshlash va uni boshqarishni kompyutersiz tasavvur qilish qiyin. XXI asr savodxon kishisi bo'lishi uchun kompyuter savodxon bo'lish, axborot texnologiyalarini

puxta egallamoq lozim. Har bir mutaxassis, u qaysi sohada ishlashdan qat'iy nazar, o'z vazifasini zamon talabi darajasida bajarishi uchun axborotni ishlab chiqaruvchi vositalar va ularni ishlatish uslubiyotini bilish va ishlash ko'nikmalarga ega bo'lishi zarur. Talabalarni ijtimoiy-iqtisodiy va ma'naviy muammolarni hal etishga safarbar qilmoq uchun tegishli axborotlarni o'z vaqtida to'plab, qayta ishlab, muayyan bir tartibga solish va zudlik bilan kishilarga etkazish kerak bo'ladi. Buning uchun jamiyatni axborotlashtirish dasturini amalga oshirish va ilg'or axborot texnologiyasini joriy etish zarurdir.

Dasturlarni mustaqil tuzishdan maqsad kompyuterga mutloq xokimlik qilish, ya'ni ish davomida yuzaga keladigan muammolarni tezroq hal etish imkonini yaratishdir. Kompyuter dasturlari sermehnat ishlarni avtomatlashtiradi, xatolarni kamaytiradi va mehnat unumdorligini oshiradi. Bundan tashqari, dasturlar tuzish juda ham mashg'ulotdir.

Dasturlarni yaratish jarayonida qo'yilgan masalaning yechish algoritmi dastlab to'g'ri ishlab chiqilishi muhim ahamiyatga ega. Shuning uchun algoritmlarni tuzish va dasturlarni ishlab chiqish bir-biri bilan chambarchas bog'liq jarayonlardir. Oliy o'quv yurtlarining informatika, axborot texnologiyalari, amaliy matematika kabi yo'nalishlarida ta'lim olayotgan talabalar algoritmlarni ishlab chiqish, dasturlar yaratish, ularni sinash, sozlash, tahlil qilish uchun bilimlarni puxta o'zlashtirishlari zarur. Bunda, ta'lim oluvchi uchun dasturlarni ishlab chiqishda asosiy va eng muhim bosqich hisoblangan algoritmlarni tuzish va shular asosida dasturlar yaratish haqida ma'lumotlarni beruvchi adabiyotlar kerak.

Ma'ruzalar matni Oliy o'quv yurtlari talabalari uchun mo'ljallab yozilgan va zamonaviy kompyuter texnologiyalarini mustaqil ravishda o'rganayotgan barcha qiziquvchilar uchun ham foydalidir.

## **1 - MAVZU: KIRISH. ALGORITMLASH FANI VA ALGORITMLASH SAN'ATI**

Algoritm tushunchasi zamonaviy matematika va informatikaning asosiy tushunchalaridan biri hisoblanadi. Algoritm termini o'rta asrlar ulug' matematigi al-Xorazmiy nomidan kelib chiqqan. XX asrning 30-yiligacha algoritm tushunchasi ko'proq matematik ma'no emas, balki metodologik ma'noni kasb etar edi. Algoritm deganda, u yoki bu masalalar sinfini yechish imkonini beruvchi aniq ifodalangan chekli qoidalar majmui tushunilgan. EHM larning paydo bo'lishi bilan algoritm tushunchasi yanada keng tarqaldi. EHM va dasturlash usullarining rivojlanishi algoritmlarni ishlab chiqish avtomatlashtirishdagi zaruriy bosqich ekanligini tushunishga yordam berdi. EHM larning paydo bo'lishi algoritmlar nazariyasining rivojlanishiga olib keldi.

Algoritmlarni tuzish – bu ijodiy ish bo'lib, ixtiyoriy zaruriy algoritmlarni tuzish uchun umumiy usullar mavjud emas, kishining ijodiy qobiliyatiga bog'liq.

Albatta, algoritmlarni aniq sxema bo'yicha tuzish zarur bo'lib qoladigan sodda hollar ham mavjud. Bunday hollarda yechilish algoritmi avval biron kim tomonidan olingan masalalarni misol keltirish mumkin. Masalan, differensial tenglamalarni sonli integrallash uchun Eyler metodi. Bu metod masalani yechish uchun umumiy holda ifodalangan algoritmdir, lekin algoritmlash ijodiy ekanligini quyidagi algoritmlar nazariyasining ba'zi bir ma'lumotlaridan ko'rish mumkin.

Agar bizdan biror algoritmlarni ishlab chiqish talab qilinsa, dastlab izlanayotgan algoritmlarni tuzish mumkinmi yo'qmi degan savolga javob izlash kerak. Chunki ba'zi hollarda algoritmlarni tuzish mumkin emasligini ko'rsatib berish mumkin. Ba'zi bir hollarda algoritmlarni tuzish mumkinligi isbotlanadi. Bunday isbot mavjud bo'lganligi bilan tuzilgan algoritmlarni amalgam oshirib bo'lmaydi yoki uning samaradorligi

talabga javob bermaydi. Shunga qaramasdan bir nechta algoritmlar bitta amaliyotga qo'llanilayotganini topish mumkin.

Boshqa hollarda algoritmni tuzish mumkinligini ham, mumkin emasligini ham isbotlab bo'lmaydi. U vaqtda algoritm tuzish jarayonida boshqa predmet sohalaridan qurilgan algoritmlardan foydalanish mumkin.

Algoritmlar sifatini baholash uchun mezonlarni ko'raylik. Mavjud mezonlar juda tahminlashgan. Masalan, algoritmni bajarishda bajaruvchining xotira uskunalari hajmi yetarli bo'lmasa, u algoritm yomon deb hisoblanadi. Boshqa mezon sifatida algoritmning bajarilishi uchun talab qilinadigan vaqtni ko'rsatish mumkin. Vaqtni baholash bajaruvchining fizik xarakteristikalari hisobga olinishi kerak. Chunki har bir operatsiya har xil o'zgaruvchilar bilan bajarilganda vaqt ham har xil bo'ladi. Bunchalik aniq ma'lumotni har bir foydalanuvchi uchun yig'ib bo'lmaganligi sababli odatda o'rtacha tezkorlik qabul qilinadi. Ketma-ket bajarilayotgan operatsiyalar sonini aniqlab, uni o'rtacha tezkorlikka ko'paytirsa, algoritm bajarilishining amalga yaqin bo'lgan vaqtini topishimiz mumkin.

Faraz qilaylik, 2 ta tahlil qilingan algoritmlardan bittasining bajarilish vaqti tezroq bo'ladi, uni xotira ishlash hajmi bo'yicha ham tahlil qilish kerak va bunday tahlillar murakkab nazariyasiga mansub bo'ladi. Shunday qilib, algoritmlar nazariyasi fani masalalarni yechishga mo'ljallangan algoritmlarni samaradorligini va murakkabligini tahlil qilish, o'zgartirish, qo'shimcha qilish va qayta ishlash natijasida yaxshilash usul va uslublarini o'rganadi.

## 2 - MAVZU: ALGORITMLASHNING MATEMATIK ASOSLARI

### *Reja*

1. *Matematik induksiya .*
2. *Yig'indi va Ko'paytmalar.*
3. *Butun qiymatli funksiyalar.*
4. *O'rin almashtirishlar va faktoriallar.*
5. *Binomial koeffitsiyentlar.*
6. *Fibonachi sonlari.*

Algoritmlarni tuzishda va ularning tahlilida ishlatiladigan ba'zi matematik belgilashlarni qarab chiqamiz.

### Matematik induksiya .

Faraz qilaylik  $P(n)$  – bu  $n$  butun son to'g'risidagi biror bir tasdiq bo'lsin. « $n(n+3)$  – juft son»  $n \geq 10$  bo'lsa, u holda  $2^n > 2^{3n}$ . Bizdan  $P(n)$  ning barcha butun musbat  $n$  sonlar uchun o'rinli ekanligini isbotlash talab qilinsin. Isbotning asosiy usuli quyidagilardan iborat:

1.  $P(1)$  o'rinli ekanligini isbotlash.
2.  $P(1), P(2), \dots, P(n)$  lar o'rinli bo'lsa, u holda  $P(n+1)$  ham o'rinli ekanligini isbotlash, bu isbot barcha butun musbat  $n$  lar uchun o'rinli bo'lishi kerak.

#### **Misolni keltiramiz.**

$$1 = 1^2 \quad 1 + 3 = 2^2 \quad 1 + 3 + 5 = 3^2 \quad 1 + 3 + 5 + 7 = 4^2 \quad (1)$$

Ularning umumiy ko'rinishda quyidagicha yozish mumkin:

$$P(n) = 1 + 3 + \dots + (2n - 1) = n^2 \quad (2).$$

Biz  $P(n)$  ning barcha musbat  $n$  lar uchun o'rinli ekanini isbotlashimiz kerak. Yuqoridagi proseduraga muvofiq:

- a).  $P(1)$  o'rinli, chunki  $1 = 1^2$
- b). agar barcha  $P(1), P(2), \dots, P(n)$  tasdiqlar o'rinli bo'lsa,  $P(n)$  uchun ham o'rinli, ya'ni (2) munosabat bajariladi.

(2) ning har ikkala tomoniga  $2n+1$  ni qo'shsak, quyidagiga ega bo'lamiz:

$$1 + 3 + 5 + \dots + (2n - 1) + 2n + 1 = n^2 + 2n + 1 = (n + 1)^2$$

Bu esa  $P(n+1)$  ning ham to'g'riligini ko'rsatadi.

Bu metodni isbotlashning algoritmik prosedurasini deb qarash mumkin. Haqiqatan ham, agar a) va b) bosqichlar amalga oshgan deb hisoblasak, quyidagi algoritmi P(n) tasdiqning ixtiyoriy butun musbat n uchun isbotini beradi.

Berilgan butun musbat n uchun P(n) ning o'rinli ekanini isbotlash algoritmi.

A1 algoritmi.

1. boshlash.
  2.  $k \rightarrow 1$  {(a)ga asosan P(1) tasdiqni isbotlang}
  3. agar  $k=n$  bo'lsa, u holda 6 ga o'ting
  4. p(k+1) uchun isbotlang ((b) ga asosan p(2), p(3), p(k) to'g'riligini isbotlang va p(k+1) uchun to'g'ri degan xulosaga keling)
  5.  $k \leftarrow k + 1$  3 ga o'ting
  6. tugash (so'ralayotgan isbot bajarildi)
- (a) va (b) bosqichlar (a1 algoritmi) shaklidagi isbotlash matematik induksiya yordamida isbotlashdir

**Yig'indi va Ko'paytmalar.**

$a_1, a_2, \dots$  - ixtiyoriy sonlar ketma-ketligi bo'lsin.  $a_1 + a_2 + \dots + a_n$  ko'rinishdagi yig'indini  $\sum_{i \leq j \leq n} a_j$  kompakt ko'rinishida yozish mumkin.

Agar n nolga yoki manfiy songa teng bo'lsa berilishiga ko'ra bu yig'indi nolga teng bo'ladi. j harfi indeks yoki yig'indining o'zgaruvchisi.

Yig'indilar chekli (j qiymatlarini chekli soni) va cheksiz bo'lishi ham mumkin. Agar  $\sum$  belgisi ostida ikki yoki undan ortiq shartlar joylashgan bo'lsa, ularning barchasi bir vaqtning o'zida bajarilish kerak.

Yig'indi uchun qisqa yozuv bo'lganidek, ko'paytma uchun ham  $\prod_{1 \leq j \leq n} a_j$  qisqa yozuv ishlatiladi.  $\prod_{1 \leq j \leq n} a_j$

belgi  $1 \leq j \leq n$  shartni qanoatlantiruvchi barcha butun j lar uchun barcha  $a_j$  lar ko'paytma lga teng deb hisoblanadi (yig'indi esa nolga teng bo'ladi).

**Butun qiymatli funksiyalar.**

Ixtiyoriy haqiqiy son uchu quyidagi belgilashlarni kiritamiz:

$\lfloor x \rfloor$  - x ga eng yoki x dan kichik bo'lgan eng katta butun son.

$\lceil x \rceil$  - x ga eng yoki x dan katta bo'lgan eng kichik butun son.

Bu funksiyalar ni ba'zida x sonining butun qismi deb yuritiladi.

Masalan:  $\lfloor \sqrt{2} \rfloor = 1$   $\lceil \sqrt{2} \rceil = 2$   $\lfloor x \rfloor = \lceil x \rceil$ .

Ixtiyoriy haqiqiy x va y sonlar uchun quyidagi Binar amalini belgilaymiz.  $X \bmod Y$  - x ni y ga bo'lgandagi qoldiqni bildiradi. Agar x va y lar butun son bo'lsa, u holda qoldiq ham butun son va x,y ga karrali bo'lsa, nol bo'ladi.

$5 \bmod 3 = 2$

$18 \bmod 3 = 0$

Agar x va y butun sonlar bo'lsa, div butun qiymatli bo'lishni bildiradi, ya'ni butun qiymatli bo'lish natijasida har doim butun bo'ladi.

$7 \div 2 = 3$

$2 \div 5 = 0$

**O'rin almashtirishlar va faktoriallar.**

n tartibli o'rin almashtirish deb, n ta turli ob'yektlarni qatorga joylashtirish operatsiyasiga aytiladi.

Masalan, a, b, c lar uchun 6 ta o'rin almashtirishlar bor. abc, bac, bca, cab, acb.

n ob'yektdan tuzish mumkin bo'lgan umumiy o'rin almashtirishlar soni

$P(n) = n(n-1)(n-2) \dots 1 = n!$

P(n) qiymatni n! deb hisoblaydilar va u quyidagicha yoziladi.

$n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n = \prod_{1 \leq k \leq n} k$

0!=1 ekanligi qabul qilingan. Butun musbat n lar uchun  $n! = (n-1)! \cdot n$  ayniyat o'rinli. 0!=1 1!=1 3!=6.

Faktoriallar juda tez o'sadi. 10!=3628800

1000! esa 2500 dan ortiq o'qli belgilardan iborat. Shunga qaramasdan kompyuterda faktorialni hisoblash uchun kam vaqt ketadi.

Dj. Stirling degan olim  $n! \approx \sqrt{2\pi n} \left(\frac{n}{e}\right)^n$  ga teng deb olgan.

Yana bir savol tug'ildi. Biz  $n!$  uchun  $n$  butun musbat bo'lgan hol uchun ta'rif berdik.  $n$  ning ratsional qiymatlari yoki  $n$  haqiqiy bo'lganda  $n!$  nimaga teng degan savol tug'iladi. Masalan,  $\left(\frac{1}{2}\right)!$  nimaga teng. Bu masalani yechish uchun butun manfiy masalalar uchun  $n!$  ni aniqlaymiz.

$$n! = 1 + 2 + \dots + n = \sum_{1 \leq k \leq n} k \quad (1)$$

Bu faktorialning analogi, lekin bu yerda biz ko'paytirish o'rniga qo'shishdan foydalanayapmiz

$$\text{Arifmetik progressiyaning yig'indisi} \quad n! = \frac{1}{2} n(n+1) \quad (2)$$

(2) ni (1) ning o'rniga ishlatish  $n!$  funksiyani  $n$  ning ixtiyoriy qiymatlari uchun aniqlash imkonini beradi.

Masalan,  $\left(\frac{1}{2}\right)! = \frac{3}{8}$ .

### **Binomial koeffitsiyentlar.**

$n$  ta ob'yektdan  $k$  ta ob'yektni jamlash bu  $n$  ta elementdan mumkin bo'lgan  $k$  ta turli elementni tanlash. Masalan, 5 ob'yektdan 3 tadan jamlash, a, b, c, d, e. abc, abd, abe, acd, ace, ade, bcd, bce, bde, cde.

$n/k$  orqali belgilangan jamlashni umumiy soni

$$\binom{n}{k} = \frac{n(n-1)\dots(n-k+1)}{k(k-1)\dots 1}$$

Masalan  $\binom{5}{3} = \frac{5 \cdot 4 \cdot 3}{3 \cdot 2 \cdot 1} = 10$ .

$\binom{n}{k}$  qiymat binomial koeffitsiyent deb aytiladi. Binomial koeffitsiyentni faktorial yordamida hisoblash

mumkin.  $\binom{n}{k} = \frac{n!}{k!(n-k)!}$

Binomial koeffitsiyentlar uchun quyidagi hossa mavjud:

$$\binom{r}{k} = \frac{r}{k} \binom{r-1}{k-1}$$

### **Fibonachi sonlari.**

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55,...

ketma-ketlikda har bir son oldingi 2 ta sonning yig'indisiga teng bo'lsa, Fibonachi sonlari deb aytiladi.

$$F_n = F_{n-1} + F_{n-2} \quad n > 0.$$

Bu ketma-ketlik Leonardo Fibonachi tomonidan taklif etilgan. Fibonachi sonlari va algoritmlar orasida o'zaro bog'liq borligi isbotlangan.

Takrorlash uchun savollar

1. Matematik induksiya haqida tushuncha bering.
2. Yig'indi va Ko'paytmalarning asosoy farqini ko'rsating.
3. Butun qiymatli funksiyalarga misol keltiring.
4. O'rin almashtirishlar va faktoriallarni hisoblashga misol ko'rsating.
5. Binomial koeffitsiyentlar - bu nima?
6. Fibonachi sonlari algoritmlarga qanday aloqasi bor?

### **3 - MAVZU: ALGORITMLAR VA ULARNING TO'LIQ TUZULISHINING BOSQICHLARI**

## *Reja*

- 1. Algoritmning ta'rifi.*
- 2. Algoritmni to'liq yaratish bosqichlarni*
- 3. Masalaning qo'yilishi.*
- 4. Modelni yaratish.*
- 5. Algoritmni ishlab chiqish.*
- 6. Algoritm to'g'riligini tekshirish.*
- 7. Algoritmni amalga oshirish.*
- 8. Algoritmni va ularning murakkabligini tahlil qilish.*
- 9. Dasturni tekshirish.*
- 10. Hujjatlashtirish.*

Algoritmning turli ta'riflari mavjud. Rasmiy ta'riflardan biri bo'yicha algoritm bu qo'yilgan masalani bir xil yechilishiga olib keluvchi aniq harakatlarning ketma-ketligi. Bu tushunchadan algoritmning quyidagi xossalari kelib chiqadi:

1. Diskretlilik – ya'ni aniqlanayotgan jarayonni qadamba-qadam ko'rinishi.
2. Ommaviylik – algoritm o'xshash masalalar turkumini yechishi kerak.
3. Tushunarlilik – algoritmda beriladigan ko'rsatmalar foydalanuvchiga tushunarli bo'lib, uning talablariga javob berishi kerak.
4. Aniqlilik – algoritmda ma'lum tartibda amallarni bajarish nazarda tutilishi kerak va bajaruvchiga joriy qadam tugatilishi bilan qaysi qadam keyingi bo'lib bajarilishi aniq ko'rsatilishi kerak.

Algoritm rasm ravishda bajariladi, bu degani bajaruvchi bajarilayotgan amallarni mazmunini anglash shart emas. Algoritm tuzish jarayoniga algoritmlashtirish deyiladi.

Algoritm tuzish jarayonida nazariy va amaliy nuqtai nazardan algoritmlash, dasturlash va EHM larni qo'llash bilan bog'liq bo'lgan bilimlar kerak. Asosiy maqsad bu masalani qo'yish, masalaning yechish algoritmini tuzish, algoritmi mashina dasturi ko'rinishida amalga oshirish va algoritmni samaradorligini ko'rsatish muammolarini o'rganish. Bu jarayonlar algoritmni to'liq yaratish tushunchasiga olib keladi va quyidagi bosqichlarni belgilaydi:

1. Masalaning qo'yilishi.
2. Modelni yaratish.
3. Algoritmni ishlab chiqish.
4. Algoritm to'g'riligini tekshirish.
5. Algoritmni amalga oshirish.
6. Algoritmni va ularning murakkabligini tahlil qilish.
7. Dasturni tekshirish.
8. Hujjatlashtirish.

### **Masala qo'yilishi**

Masalani yechishdan oldin, uni berilishini aniq shakllantirib olish zarur. Bu jarayon to'g'ri savollarni aniqlash bo'lib, savollar quyidagicha bo'lishi mumkin:

- 1.1. Dastlabki berilgan masala shartlarida hamma iboralar tushunarlimi?
- 1.2. Nima berilgan?
- 1.3. Nimani topish kerak?
- 1.4. Yechimni qanday ta'riflash kerak?
- 1.5. Qaysi berilganlar yetarli emas va hammasi kerakmi?
- 1.6. Qanaqa mumkinliklar qabul qilingan?

Albatta, bulardan tashqari boshqa savollarni ham ishlatish mumkin, yoki ayrim savollarni bir necha bor takror ishlatishga to'g'ri keladi.

### **Modelni yaratish**

Akademik A. N. Tixonov fikri bo'yicha matematik modellashtirish dunyoni bilish va o'rganishda kuchli qurollardan (vositalardan) biridir.

Uning ta'rifi bo'yicha matematik model tashqi dunyoning xodisalar turkumini matematik belgilar yordamida taxminiy tavsifi.

Xodisani tavsiflash uchun uning muhim xususiyatlarini, qonuniyliklarini, ichki aloqalarini, ayrim xossalarning ahamiyatini aniqlash zarur. Eng muhim faktorlari aniqlanganda, ahamiyatlari kamroq bo'lganlarini hisobdan chiqarish mumkin. Umuman, modelni tanlash fandan ko'ra, ko'proq san'at ishi deb hisoblanadi, yahshi tuzilgan modellarni o'rganish esa – modellashtirishda tajriba orttirishning eng yahshi usuli. Modelni yaratishda quyidagi savollarni aniqlash maqsadga muvofiq:

- 2.1. Masalani yechish uchun qaysi matematik struktura ko'proq mos keladi?
- 2.2. O'xshash masalaning yechimi bormi?
- 2.3. Masalaning barcha muhim ma'lumotlari matematik ob'yektlar orqali tavsiflanadimi?
- 2.4. Izlanayotgan natija biron bir matematik o'lchamga mos keladimi?
- 2.5. Modelning ob'yektlari orasidagi bog'lanishlar aniqlanganmi?
- 2.6. Tuzilgan model bilan ishlash qulaymi?

### **Algoritmni ishlab chiqish**

Algoritmshirish jarayoni uslublari bo'yicha matematik modellarni tuzish jarayoniga juda yaqin. Har bir algoritmni ishlab chiqish bevosita o'ziga xos yondashishni talab qilishiga qaramasdan, bu faoliyatni umumiy uslub va bosqichlari ham mavjud. Ba'zan dasturlarni tezroq yozib boshlashga hohish paydo bo'ladi. Lekin bu xatoli, chunki aynan algoritmni ishlab chiqish bosqichiga va uning to'g'riligiga masalaning to'liq yechimi bog'liqdir. Algoritmni tuzish turli xil uslublari mavjud.

### **Algoritmni to'g'riligini tekshirish**

Dastur to'g'riligini isbotlashning eng keng tarqalgan turi – bu uni testlardan o'tkazishdir.

Algoritmni tekshirishda nazoratchi boshlang'ich ma'lumotlarni majmui algoritmik test deb nomlanadi.

To'g'ri deb shunday algoritmga aytiladiki, u masalaning qo'yilishida talab qilinadigan natijani har qanday ruxsat etilgan boshlang'ich ma'lumotlar bilan ham shakllantirib biladi. Odatda, dastur bergan natijalar ma'lum bo'lgan yoki qo'lda hisoblangan ma'lumotlar bilan taqqoslanadi, va ular to'g'riligi aniqlansa dastur to'g'ri ishlaydi degan hulosaga kelish mumkin. Ammo bu usul bilan foydalanuvchini hamma shubhalardan xalos qilib bo'lmaydi, ya'ni dastur ishlamaydigan hamma holatlarni hisobga olib bo'lmaydi.

Gudman va Xidetniyemi [2] lar tomonidan algoritm to'g'riligini isbotlash uchun quyidagi uslubiyat taklif qilingan.

Algoritm 0 dan  $m$  gacha bo'lgan qadamlar ketma-ketligi ko'rinishida tavsiflangan deb tahmin qilaylik. Har bir qadam uchun qandaydir asoslanishni taklif etamiz. Xususan, qadamdan oldin va keyin ishlaydigan shartlar haqida lemma kerak bo'lishi mumkin. Shu bilan birgalikda, algoritm chekliligining isbotini ham taklif etamiz, va hamma ruxsat etilgan kiritish ma'lumotlarini tekshirib, hamma mumkin bo'lgan chiqarish ma'lumotlarni olamiz. Algoritmni to'g'riligi bilan samaradorligi o'rtasida hech qanday aloqa yo'qligini ta'kidlab o'tamiz. Aslida hamma talablarga bir xil yahshi javob beradigan algoritm kamdan-kam ishlab chiqiladi.

### **Algoritmni amalga oshirish**

Algoritmni amalga oshirish deganda, EHM uchun dasturni yozish deb tushuniladi. Buning uchun quyidagi savollarga javob berish kerak:

- 5.1. Asosiy o'zgaruvchilarni aniqlash.
- 5.2. O'zgaruvchilarning turlarini aniqlash.
- 5.3. Nechta massiv yoki fayllar va qanday kattalikda ular kerak bo'ladi?
- 5.4. Bog'lanilgan ro'yhatlardan foydalanish ma'nolimi?
- 5.5. Qanday dasturiy qismlar kerak bo'lishi mumkin (tayyor bo'lsa ham)?
- 5.6. Qaysi dasturlash tilini tanlash?

Dastur yozish yoki tuzishning hilma-hil usillari va uslublari mavjud.

### **Algoritmni va uning murakkabligini tahlil qilish**

Algoritmni tahlil qilishdan maqsad – algoritmgaga ma’lumotlarni aniq muvaffaqiyatli qayta ishlash uchun kerak bo’ladigan xotira hajmi va ishlash vaqtining baholari va chegaralarini olish. Bir masalani yechadigan ikki algoritmni taqqoslash uchun qandaydirsonli mezon topish kerak.

Faraz qilaylik, A – qandaydir bir turkumdagi masalalarni yechadigan algoritm, n – esa shu turkumdagi alohida bir masalaning kattaligi. Umumiy holda, n – oddiy skalyar yoki massiv yoki kiritiladigan ketma – ketlikning uzunligi bo’lishi mumkin.  $f_A(n)$  - n kattalikdagi ixtiyoriy masalani yechadigan algoritm A bajarish kerak bo’lgan asosiy amallarni (qo’shish, ayirish, taqqoslash,...) yuqori chegarasini beradigan ishchi funksiya. Algoritmning sifatini baholash uchun quyidagi mezonni ishlatamiz.

Agar  $f_A(n)$  o’sish tartibi n dan bog’liq bo’lgan polinomdan katta bo’lmasa, A algoritm polinomial deb aytiladi, aks holda algoritm A eksponensial hisoblanadi.

Shular bilan birgalikda tahlil jarayonida ko’p matematik fanlarda standart bo’lgan iboralar ishlatiladi.

$f_A(n)$  funksiya  $O[g(n)]$  deb belgilanadi, va  $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = const \neq 0$  bo’lganda, uni tartibi katta n lar uchun  $g(n)$  deb qabul qilinadi. Demak  $f(n) = O[g(n)]$ .

$f_A(n)$  funksiyasi  $o[z(n)]$  deb katta n lar uchun belgilanadi, va unda  $\lim_{n \rightarrow \infty} \frac{h(n)}{z(n)} = 0$  sharti bajariladi.

Bu begilar “katta O” va “kichik o” deb nomlanadi. Agar  $f(n) = O[g(n)]$  bo’lsa, ikkala funksiya ham  $n \rightarrow \infty$  bo’lganda bir xil tezlikda o’sadi.

Agar  $f(n) = O[g(n)]$  bo’lsa, unda  $g(n)$ ,  $f(n)$  nisbatan ancha tez o’sadi.

Demak,  $P_k(n)$  - qandaydir n o’zgaruvchidan bog’liq va k darajadagi polinom uchun  $f_A(n) = O[P_k(n)]$  yoki  $f_A(n) = oP_k(n)$  bo’lganda algoritm polynomial hisoblanadi, aks holda algoritm eksponensial.

Eksponensial algoritm yahshi ishlaymaydigan deb hisoblanadi. Agar algoritmlar eksponensial bo’lsa, ular orasida eng samaralisini topish kerak, n kattalikdagi masalani  $O(2^n)$  qadamda yechadigan algoritm  $O(n!)$  yoki  $O(n^n)$  qadamda masalani yechadigan algoritmdan afzalroq.

### **Dasturni tekshirish**

Biz dasturni har bir qismini tekshiradigan kirituvchi ma’lumotlar to’plamini tanlashimiz kerak. Ko’p murakkab algoritmlarni matematik tomondan tadqiq qilish yoki juda qiyin yoki mumkin emas. Bunday holatlarda algoritmni faoliyat jarayonida va qiyinligi bo’yicha tekshiradi. Bundan tashqari dasturlarni hisoblash imkoniyatlarini aniqlash uchun ham testlash maqsadga muvofiq. Ko’p dasturlar qandaydir kiritiladigan ma’lumotlar bilan yahshi ishlasa, boshqalari bilan yomon ishlaydi. “Yahshi” lardan “yomon” larga o’tish “mayin” bo’lish kerak. Testlash uchun ma’lumotlar dasturning qiyinligiga, mavjud vaqt resurslariga, kiritish-chiqarishsoniga bog’liq holda tanlanadi. Bu yerda analitik va eksperimental tahlil bir-birini to’ldiradi.

### **Hujjatlashtirish**

O’zingiz yozmagan dastur kodini o’qish juda qiyin. Bu muammoni hujjatlashtirish yordamida yechsa bo’ladi. Hujjatlashtirish o’z ichiga hamma yordamchi ma’lumotlarni oladi va dasturda nima bajarilishini tushuntirib beradi, xususan, blok-sxemalardagi boshqarishni uzatish, berilganlarni kiritish-chiqarish shaklini batafsil tavsif qilish, siklning parametrlari, yordamchi local va global proseduralarni bajarilishi va boshqalar.

Hujjatlashtirishning eng asosiy qoidasi bu “boshqalar yozgan dasturlarni qanday ko’rishni istasangiz, o’zingiz ham dasturni shunday ko’rinishda rasmiylashtiring”.



## Takrorlash ucun savollar

1. *Algoritmning qaysi ta'riflarinin bilasiz?*
2. *Algoritmni to'liq yaratish bosqichlarini aytib o'ting*
3. *Masalani qo'yishda va modelni yaratishdagi savollarni qanday aniqlash kerak?*
5. *Algoritmni va ularning murakkabligini tahlil qilishda nimalarga e'tibor berish kerak?*

## 4 - MAVZU: ALGORITMLARNI TAVSIFLASH TILI HAQIDA KELISHUV

### *Reja*

1. *Algoritmning umumiy ko'rinish*
2. *Tarmoqlanuvchi yoki shartli buyruqlar*
3. *Tanlash buyruqlari.*
4. *Takrorlash buyruqlari.*

Algoritmni tavsiflash usullari haqida aytadigan bo'lsak, ular so'zlar bilan, jadvallar bilan, grafik yoki blok-sxemalar va maxsus sun'iy tilda berilishi mumkin.

So'zli tavsiflashda tabiiy til va matematik belgilar elementlari ishlatiladi. Jadvalli berilishda algoritm jadvallar va hisoblash formulalari shaklida ko'rsatiladi. Grafikli, blok-sxemali va graflar yordamida berilishda algoritm geometrik figuralar va bloklar yordamida tavsiflanadi.

Algoritmik til – bu algoritm va ular bajarilishini bir qolirda va aniq yozish uchun belgilar va qoidalar tizimi. Algoritmik til so'zlarni tuzishga xizmat qiladigan o'z lug'atiga ega bo'ladi. So'zlar algoritmni buyruqlarini yozish uchun ishlatiladi. Bundan tashqari tilning xizmat so'zlari ham bo'ladi, ularning berilishi va ma'nosi aniq belgilangan.

So'zli algoritmik tillarga misol bo'lib "Informatika" kursidagi maktab algoritm tili xizmat qilishi mumkin. Bu til yordamida ixtiyoriy algoritmni o'qishva tahlilga qulay ko'rinishda, hamda unga qandaydir qo'shimcha buyruqlar va konstruksiyalar kiritish imkoni bilan yozish mumkin.

### **Algoritmning umumiy ko'rinishi va shu tilning asosiy buyruqlari.**

#### **1. Algoritmning umumiy ko'rinishi:**

Alg nomi (argumentlar va natijalar ruyhati).

Arg ruyhat.

Nat ruyhat.

Bosh

Buyruqlar ruyhati.

Tamom.

#### **2. Tarmoqlanuvchi yoki shartli buyruqlar.**

Agar shart.

Unda ruyhat 1.

Aks holda ruyhat 2.

Tamom.

#### **3. Tanlash buyruqlari.**

Tanlash

Shart 1: ruyhat 1.

Shart 2: ruyhat 2.

.....

Shart N: ruyhat N.

Tamom.

#### **4. Takrorlash buyruqlari.**

1. toki shart.

Sikl bosh ruyhat.

Sikl tug.

2. Takror ruyhat to shart .

3.  $i=n$  dan  $m$  gacha

sikl bosh ruyhat

sikl tug.

Bu tilda yozilgan algoritmlarni yuqori darajali dasturlash tiliga bevosita o'tkazish oson. Algoritmni tuzishda va tahlil qilishda bu yerda faqatgina qabul qilingan algoritmik tildagi konstruksiyaga mos buyruqlarni bajarish uchun kerak bo'ladigan vaqt va xotira muhim.

Takrorlash ucun savollar

1. *Algoritmni tavsiflash uchun qaysi tillardan foydalansa bo'ladi?*
2. *Asosiy konstruksiyalarni blok-sxema yordamida ifodalang.*
3. *Asosiy konstruksiyalarni Paskal dasturlash tilida ifodalang.*
4. *Asosiy konstruksiyalarni C++ tilida ifodalang.*

## 5 - MAVZU: ALGORITMLAR VA ULARNING QIYINLIGI

*Reja*

1. *Algoritmni baholash mezonlari.*
2. *Algoritmni vaqt qiyinligi bo'yicha optimallashtirish.*
3. *Algoritmni hajmiy qiyinligi bo'yicha optimallashtirish.*

Algoritmlarni baholash uchun ko'pgina mezonlar mavjud. Odatda kirituvchi berilganlarni ko'payishida masalani yechish uchun kerak bo'ladigan vaqt va xotira hajmlarini o'sish tartibini aniqlash muammosi qo'yiladi. Har bir aniq masala bilan kiritiladigan berilganlarni miqdorini aniqlovchi qandaydir sonni bog'lash zarur. Bunday son masalaning kattaligi deb qabul qilinadi. Masalan, ikkita matritsani ko'paytirish masalasining o'lchami bo'lib, matritsalar kattaligiga xizmat qilishi mumkin. Graflar haqidagi masalada o'lcham sifatida graf shohlarining soni bo'lishi mumkin.

Algoritm sarflanayotgan vaqt masalaning o'lchami funksiyasi sifatida algoritmni vaqt bo'yicha qiyinligi deb nomlanadi. Bunday funksiyaga masalaning kattaligi oshganda limit ostidagi o'zgarish asimptotik qiyinlik deb aytiladi.

Shunga o'xshab, hajmiy qiyinlik va asimptotik hajmiy qiyinlikni aniqlash mumkin.

Aynan algoritmning asimptotik qiyinligi natijada shu algoritm yordamida yechiladigan masalarni kattaligini aniqlaydi. Agar algoritm  $n$  kattalikdagi kirishlarni  $C \cdot n^2$  vaqtda qayta ishlasa ( $c$ -const), unda algoritmning vaqt bo'yicha qiyinligi  $O(n^2)$  teng deb hisoblanadi, va  $n$  tartibda deb aytiladi.

Hisoblash mashinalar tezligi oshishiga qaramasdan, ular yordamida yechilayotgan masalalar kattaligini oshishini algoritm qiyinligini tahlil orqali aniqlaydi.

Faraz qilaylik,  $A1, A2, \dots, A5$  nomli 5 ta algoritm quyidagi vaqtli qiyinliklar bilan berilgan.

Algoritm	Vaqtli qiyinlik
A1	$N$
A2	$N \log_2 n$

A3	$N^2$
A4	$N^3$
A5	$2^n$

Bu yerda vaqtli qiyinlik – bu  $n$  kattalikdagi kirishlarni qayta ishlash uchun kerak bo'ladigan vaqt birliklar soni. Masalan, vaqt birligini 1 millisekund deb qabul qilaylik.

Bunda A1 algoritmi bir sekundda 1000 kattalikdagi kirishni qayta ishlash mumkin, A5 algoritmi esa kirish kattaligidagina 9 dan oshirib bilmaydi.

Keyingi jadval 1 sekundda, 1 minutda, 1 soatda 5 ta algoritmlarni har birining yordamida yechiladigan masalaning kattaligi keltirilgan.

Algoritm	Vaqtli qiyinlik	Masalaning maksimal o'lchami		
		1 sek	1 min	1 soat
A1	$N$	1000	60*100	3,6*10 <sup>6</sup>
A2	$N \log_2 n$	140	4893	2*10 <sup>4</sup>
A3	$N^2$	31	244	1897
A4	$N^3$	10	39	153
A5	$2^n$	9	15	21

Faraz qilaylik, keyingi avlod hisoblash mashinalari birinchi jadvalga nisbatano'n barobar tezligi oshadi. Keyingi jadvalda shunday oshishga nisbatan yechiladigan masalalar kattaligining oshishi ko'rsatilgan.

Algoritm	Vaqtli qiyinlik	Masalaning maksimal kattaligi		
		1 sek	1 min	1 soat
A1	$N$	$s_1$	$10s_1$	$\approx 10$
A2	$N \log_2 n$	$s_2$	$10s_2$	$\approx 10$
A3	$N^2$	$s_3$	$3,16s_3$	$\approx 3$
A4	$N^3$	$s_4$	$2,15s_4$	$\approx 2$
A5	$2^n$	$s_5$	$s_5 + 3,3$	$\approx 10/3$

yerda

algoritm uchun tezlikni 10 barobar oshishi masalaning kattaligining uchga oshishiga olib keladi. A3 algoritmi esa kattalik uch barobardan ziyod oshadi. Endi, tezlik oshishining o'rniga algoritmi kiruvchi

Bu  
A5

berilganlarning hajmini oshishini ko'ramiz. Birinchi jadval bo'yicha taqqoslash asosi sifatida 1 min.ni qabul qilsak, A4 algoritmi o'rniga A3 ni qo'llaganimizda, masalaning kattaligi 6 barobar oshadi, A4 algoritmi o'rniga A2 ni qo'llaganda esa 125 barobar oshirilishga erishamiz. Agar taqqoslash asosi sifatida 1 soatni qabul qilsak, natijalar yanada ham muhimlashadi.

Algoritm va uning qiyinligini batafsilroq muhokama qilish uchun biz algoritmi amalga oshirish uchun qo'llaniladigan hisoblash qurilmalarning modelini va hisoblashning elementar qadamini aniqlashimiz zarur. Afsus-ki, sharoitlarga mos keladigan modelni o'zi yo'q. Eng katta qiyinchilikni mashina so'zlarining kattaligi tug'diradi. Masalan, agar mashina so'zi ixtiyoriy uzunlikda butun son shaklini qabul qilsa, unda butun masalaning kodi mashina so'zi ko'rinishdagi bir son bo'lishi mumkin. Lekin mashina so'zining uzunligi cheklangan bo'lsa, unda masalaning kattaligi kamroq bo'lganda muammolar yechilsa ham, oddiy katta sonlarni xotirashda qiyinchiliklar tug'ilishi mumkin.

#### Takrorlash uchun savollar

1. *Algoritmi baholash mezonlari nima bilan farqlanadi?*
2. *Algoritmi vaqt qiyinligini qanday hisoblash kerak?*
3. *Algoritmi qaysi mezon bo'yicha optimallashtirish samarali?*

## 6 - MAVZU: ALGORITMLARNI ISHLAB CHIQISH USLUBLARI

### *Reja*

1. *Algoritmlarni konstruksiyalash*
2. *Algoritmlarni ekvivalent qayta ishlash.*
3. *Toraytiruvchi o'zgartirishlar.*
4. *Formal usulni matematikaga bog'liq bo'lmagan muammoga qo'llash.*

Algoritmlarni yaratish ijobiy ish, shuning uchun ixtiyoriy zarur algoritmlarni tuzish imkonini beradigan bir umumiy usul mavjud emas. Lekin algoritmlarni ishlab chiqishni asoslangan oddiy sxemalarini beradigan ko'pgina algoritmlashtirish nazariyalari bor. Bunday sxemalar va yangi algoritmlarni paydo qilishning o'rtasida qattai bog'liqlik kuzatiladi. Tez uchraydigan va ko'p foydalaniladigan usullarni quyidagicha ajratib olish mumkin:

1. **Algoritmlarni konstruksiyalash.** Bu usulda yangi algoritmi mavjud algoritmlardan tarkibiy qismlar sifatida foydalanib, bir-biriga moslab bir butunlik hosil qilish yo'li bilan ishlab chiqiladi.
2. **Algoritmlarni ekvivalent qayta ishlash.** Ikki algoritmi ekvivalent hisoblanishi uchun quyidagi shartlar bajarilish kerak:
  - Bittasi uchun mumkin bo'lgan dastlabki berilganlar varianti, ikkinchisi uchun ham mumkin bo'lishi kerak.

- Bir algoritmni qandaydir dastlabki ma'lumotga qo'llanilishi, ikkinchi algoritmni ham shu berilganga qo'llanilishiga kafolat beradi.
- Bir xil dastlabki berilgan ma'lumot uchun ikkala algoritm ham bir xil natija berishi. Lekin bu algoritmni ikki xil shakllarini ekvivalent deb nomlash noto'g'ridir.

Shunday qilib, algoritmni ekvivalent qayta ishlash deb, natijada dastlabki algoritmga ekvivalent algoritmni paydo qiladigan o'zgartirilishlarga aytiladi.

Misol tariqasida, algoritmni bir tildan boshqa tilga o'tkazishni keltirish mumkin. Shu bilan birgalikda algoritmni ekvivalent qayta ishlash usuli bilan keskin o'zgartirish mumkin, lekin bu holda asosiy e'tiborni dastlabki algoritmga nisbatan yahshi algoritmni yaratishga berish kerak.

**3, Toraytiruvchi o'zgartirishlar.** Bunday o'zgartirishlar natijasida dastlabki algoritmlar yechish kerak bo'lgan masalalarning xususiy holati yechimi algoritmlari ishlab chiqiladi. Odatda, bu usulda ekvivalent qayta ishlash jarayonida algoritmni ixchamlashtirish maqsadda foydalaniladi.

**4. Formal usulni matematikaga bog'liq bo'lmagan muammoga qo'llash.** Buyerda matematik muammo matematik ko'rinishga o'tkazilib, uning algoritmini ishlab chiqishga uriniladi. Agar o'xshash matematik masala yechimining algoritmi ma'lum bo'lsa, undan foydalaniladi.

Takrorlash ucun savollar

1. Har bir usul bo'yicha algoritm tuzishga misol ko'rsating.
2. Algoritmni ishlab chiqish uchun yana qanday usullarni bilasiz?

## 7 - MAVZU: MAKSIMUMNI TOPISH MASALASI

### *Reja*

1. *Masalaning qo'yilishi.*
2. *So'zli algoritmni ishlab chiqish*
3. *Algoritmni tahlil qilish*

Yuqorida orttirilgan bilimlar yordamida bir tipik masalani yechamiz:

### Masalaning qo'yilishi.

$x_1, x_2, \dots, x_n$  berilgan elementlar bo'yicha  $m$  va  $j$  larni shunday topingki,  $m = \max x_k \{1 \leq k \leq n\} = x_j$  bo'lsin. Bu yerda  $j$  mumkin bo'lganicha maksimal bo'lsin.

### So'zli algoritm

1. Boshlanish.
2.  $j := n; k := n-1; m := xn;$
3. agar  $k := 0$  unda 7 o'ting.
4. agar  $xk \leq m$  unda 6 o'ting.
5.  $j := k; m := xk;$
6.  $k := k-1; 3$  o'ting;
7. tamom.

Algoritm sodda va analizga muhtoj emas deb hisoblanadi. Lekin shu misolda murakkab algoritmn qanday tahlil qilish kerakligini ko'rsatish mumkin. Algoritm tahlili dasturlash uchun juda muhim.

Biz faqatgina bu algoritmni bajarish uchun kerak bo'ladigan vaqtni tahlil qilamiz. Buning uchun har bir qadam necha marta bajarilishini hisoblaymiz:

Qadam raqami	Necha marta bajarilishi
--------------	-------------------------

2	1
3	n
4	n-1
5	A
6	n-1

Har bir qadam necha marta bajarilishini bilgan holda, kompyuterga masalani bajarish uchun qancha vaqt kerakligini hisoblab chiqish mumkin.

Jadvalda A dan tashqari hamma qiymatlar ma'lum, A – bu joriy maksimum qiymatini necha marta o'zgartirish kerakligini ko'rsatkichi. Taxlilimiz to'liq bo'lishi uchun A ni ko'rib chiqamiz.

Tahlilning maqsadi A uchun min va max qiymatlarni topish.

1) Min A = 0,

bu holat

$$x_n = \max x_k \{1 \leq k \leq n\}$$

bo'lganda kuzatiladi.

2) Max A=n-1;

bu qiymatga

$$x_1 > x_2 > \dots > x_n$$

holatida erishiladi.

Shunday qilib A ning tahlili 0 va n-1 larning o'rta arifmetik qiymati va o'rta kvadratini chetlanishini va usullari yordamida topish masalasiga olib keladi.

Takrorlash ucun savollar

1. Masala quyilishida qaysi o'zgaruvchilar aniqlandi?
2. Algoritm qanday konstruksiyalar qatnashgan?
3. Aniqlangan noma'lum qiymat nechanchi qadamda bajariladi?
4. Algoritm tahlilini yakunga yetkazish ucun qanday usullarni qo'llash kerak?

## 8 - MAVZU: EVKLID ALGORITMI

*Reja*

1. *Masala qo'yilishi.*
2. *Algoritmni tuzish*
3. *Algoritm tahlili*
4. *Algoritm optimallashtirish*
5. *Algoritmni amalga oshirish*

### Masala qo'yilishi

Ikkita butun musbat  $m$  va  $n$  sonlar berilgan. Ularning umumiy bo'luvchisini topish talab qilinadi. Ya'ni, eng katta butun musbat son topish kerakki, unga  $m$  va  $n$  ni bo'lganda butun son chiqsin.

### Algoritmni tuzish

1. Boshlash;
2.  $m$  ni  $n$  ga bo'lamiz, qoldiq  $r$  ga teng bo'lsin;
3. Agar  $r=0$  unda  $n$ -natija; 5 o'ting;
4.  $m:=n$ ;  $n:=r$ ; 2 o'ting;
5. tamom.

### Algoritm tahlili

Shu algoritmni tadqiq qilib ko'raylik.  $m=119$ ,  $n=544$  deb qabul qilaylik. Ikkinchi qadamdan boshlaymiz. Algoritmga binoan bo'lish natijasini nolga teng deb hisoblaymiz va  $r$  ga  $119$  ni ta'minlaymiz, keyin 3-qadamga o'tamiz.  $R$  nolga teng bo'lmaganligi uchun, hech nima qilmaymiz va 4-qadamga o'tamiz. Bu yerda  $m$  ga  $544$  ni,  $n$  ga  $119$  ni ta'minlaymiz. Umuman, ravshan bo'ldiki,  $m < n$  bo'lsa, 2-qadamda  $m$  va  $n$  larga nisbatan hech qanday amallar bajarilmaydi, algoritm esa  $m$  va  $n$  o'zgaruvchilar qiymatlari o'rin almashishiga olib keladi.

### Algoritm optimallashtirish

Algoritmni optimallashtirish uchun unga quyidagi qadamni qo'shamiz:

1.1. agar  $m < n$   $t:=n$ ;  $n:=m$ ;  $m:=t$ ;

Endi 2-qadamga kelsak,  $544:119=4,68/119$ .  $r$  ga  $68$  ni ta'minlaymiz. 3-qadam ishlamaydi. 4-qadamda  $n=68$ ,  $m=544$ ,  $r=68$ . Keyingi sikllarda ( $r=51$ ,  $m=68$ ,  $n=51$ ), keyin ( $r=17$ ,  $m=51$ ,  $n=17$ ) va  $51/17$ , ya'ni  $r=0$ .

Shunday qilib, algoritm sikli 3- qadamda tugadi va  $544$  va  $119$  larning umumiy bo'luvchisi  $17$  ga teng bo'ldi.

Bu algoritm umumiy bo'luvchini topish uchun yagona emas. Bunday algoritmni topish uchun Dj. Steynning binar algoritmi, yoki V, xorrisning algoritmidan foydalaniladi.

### Algoritmni amalga oshirish

Shu algoritmni kompyuterda amalga oshirish uchun quyidagi Paskal dasturini keltirish mumkin:

Program evklid;

Var a, b, nod, r, x, y: integer;

Begin read (a, b);

if  $a > b$  then begin  $x:=a$ ;  $y:=b$ ; end;

else begin  $x:=b$ ;  $b:=a$ ; end;

if  $(x > 0)$  and  $(y >= 0)$  then begin while  $y <> 0$  do

begin  $r:=x \bmod y$ ;  $x:=y$ ;  $y:=r$ ; end;

Nod:=x; write (nod);

end; else write ('berilganlarda xato');

end.

Takrorlash ucun savollar

1. Masala qo'yilishidagi o'zgaruvchilarni aniqlang.
2. Algoritm optimallashtirish uchun qanday qadam qo'shildi?
3. Algoritm qaysi dasturlash tilida amalga oshirildi?

## 8 MAVZU: TASVIRLARNI TANISH MASALASI

### *Reja*

1. Masala qo'yilishi.
2. Algoritmni tuzish
3. Algoritm tahlili

#### 4. Algoritm optimallashtirish

##### Masala qo'yilishi

Etalon bilan taqqoslash muammosining bir o'lchamli holatida tasvirlarni tanish masalalaridan quyidagisini ko'ramiz. Kirish ma'lumoti sifatida  $n$  ta haqiqiy sonlardan iborat  $X$  vektor berilgan. Chiqishda shu vektorning barcha uzluksiz qism vektorlari orasida maksimal elementlar yig'indini hosil qilish kerak.

Masalan, kirish vektori

31	-41	59	26	-53	58	97	-93	-23	84
			→3				→7		

bo'lsa, unda chiqishda  $X[3..7]$  vektorning 187 qiymatli yig'indisini hosil qilamiz. Bu yerda, agar kirish vektorida hamma sonlar musbat bo'lsa, masala osonlashadi; maksimal qism-vektor sifatida kirish vektorning o'zi xizmat qiladi. Agar  $X$  vektorda manfiy sonlar ham bo'lsa, masala qiyinlashadi.

##### Algoritmni tuzish

Masalani yechish uchun, barcha elementlari manfiy bo'lgan vektorda maksimal yig'indiga ega bo'lgan vektor-qismni bo'sh vektor, ya'ni elementlar yig'indisi nolga teng bo'lgan vektorni qabul qilish shartini kiritamiz.

Eng oddiy variantda algoritm  $1 \leq L \leq U \leq N$  shartini qanoatlantiruvchi barcha  $L$  va  $U$  butun sonlar juftliklari bo'yicha  $X[L..U]$  vektorlari elementlari yig'indilarini hisoblab chiqadi va har qadamda topilgan yig'indi shungacha topilgan yig'indidan kattaligi tekshiriladi.

Psevdotilda dastur quyidagicha bo'ladi:

Maxsum:=0,0;

For L:=1 to N do

For U:=L to N do

Begin Sum:=0,0;

For i:=L to U do Sum:=Sum+x[i];

Maxsum:=max(maxsum, sum);

End.

##### Algoritm tahlili

Dasturning jiddiy kamchiligi – sekin ishlashi. 1990 yildagi o'rta tezlikka ega bo'lgan kompyuterlarda (286)  $N=1000$  bo'lganda 1 soat,  $N=10000$  bo'lganda 39 soat vaqtda bajarilgan. Bunday tezlikdagi dasturni qo'llab bo'lmaydi.

Algoritm samaradorligini intuitiv baholab ko'raylik. O-yozuvdan foydalanamiz. Eng tashqi siklning operatorlari aniq  $N$  marta bajariladi, o'rta siklning operatorlari tashqi siklning har bir qadami bo'yicha bajarilishi  $N$  dan oshmaydi. Demak, o'rta siklda bajarilayotgan 4 ta satr  $O(N^2)$  marta qiyinlik bilan baholanadi. Shu 4 ta satrlarda joylashgan sikl bajarilish soni ham  $N$  dan oshmaydi va  $O(N)$  bilan baholanadi. Baholarni ko'paytirish natijasida algoritmni umumiy bahosi  $N^3$  proporsionalligini aniqlaymiz.

“O-yozuv” usulning kamchiligi shundaki – konkret berilganlar uchun dastur bajarilishiga aniq sarflanayotgan vaqtni hisoblab bilmaymiz, faqatgina qadamlar bajarilish soni  $O(N^3)$  bo'lganini bildik.



Lekin bu usul bilan tahlil qilish qulay, va berilgan amaliy masala uchun dasturni samaradorligini aniqlaydigan dastlabki hisoblashlar uchun algoritmnining ishlash vaqtini asimptotik bahosini beradi.

Shu tahlil yordamida quyidagi algoritm yuqoridagi masalani  $O(N^2)$  qadamlar bilan yechimini ko'rsatamiz.

### **Algoritm optimallashtirish**

Bu algoritmda  $X[L..U]$  vektorning elementlar yig'indisi birinchi algoritmdagidek  $(U-L+1)$  qadamda emas, balki aniq sonli qadamlar bilan topiladi.

Yig'indini tez hisoblanishi  $X[L..U]$  vektorning elementlar yig'indisi,  $X[L..U-1]$  vektorning yig'indisiga bog'liqligiga asoslangan. Algoritm ko'rinishi quyidagicha:

```
Maxsum:=0,0;
For L:=1 to N do
  Begin sum:=0,0;
  For U:=L to N do sum:=sum+x[U];
  Maxsum:=max(maxsum, sum);
End.
```

Birinchi siklning ichidagi operatorlar  $N$  marta, ikkinchi siklning ichidagi tashqi siklning har bir qadami uchun  $N$  martadan ko'p bo'lmagan qadamlar bilan bajariladi. Demak, algoritm ishlashining umumiy vaqti  $O(N^2)$ .

Shunday qilib algoritm ishlash vaqti samaradorligi bo'yicha yaxshilandi.

Takrorlash uchun savollar

1. Masala qo'yilishidagi o'zgaruvchilarni aniqlang.
2. Algoritm optimallashtirish uchun qanday qadam qo'shildi?
3. Algoritm qaysi dasturlash tilida amalga oshirildi?

## **10 - MAVZU: KOMMIVOYAJER MASALASI**

### ***Reja***

1. Masala qo'yilishi.
2. Evristik algoritmlar.
3. GTS algoritmini tuzish
4. Algoritmni baholash

**Masala qo'yilishi.** Djek – kompyuterlar sotish bo'yicha agent (kommivoyajer), uning qaramog'ida 20 ta shahar bor. ishlayotgan kompaniya yo'l harajatlarning 50% ni to'laydi. Djek uning qaramog'ida bo'lgan har ikki shahar orasida yo'l harajatini hisoblab chiqqan. Masala yo'l harajatlarini kamaytirishdan iborat.

Dastlabki ma'lumotlar Djek tasarrufidagi shaharlar ruyhati va narxlar matrisasi ko'rinishida berilgan. Bu yerda matrisa  $i$  shahardan  $j$  shaharga borish narxiga teng bo'lgan  $c(i,j)$  elementlardan tashkil topgan ikki o'lchamli massiv. Shaharlar soni 20 ta bo'lsa, matrisa -  $20 \times 20$  bo'ladi.

Biz Djekga yo'l harajatlarini kamaytirishga yordam berishimiz kerak. Djekning marshruti o'zi yashagan shahardan boshlanib, qolgan hamma shaharlarni bir martadan o'tib, yana o'z shahriga qaytib kelishi kerak. Demak, biz tuzayotgan ruyhatda har bir shahar faqat bir marta uchrashi kerak, Lekin Djek yashagan shahar ikki marta uchrab, ruyhatning birinchi va oxirgi elementlari bo'ladi. Undan tashqari, ruyhatdagi shaharlar tartibi Djekning marshrutini belgilaydi. Ruyhatdagi ikkita oxirgi shaharlar orasidagi yo'l narxi – bu butun marshrut narxi deb hisoblanadi. Demak, agar biz Djekga eng kichik narxdagi ruyhatni tuzib bersak, masalani yechgan bo'lamiz.

### Evristik algoritmlar.

Evristika yoki evristik algoritim – algoritim deb ta’riflanishi uchun quyidagi hususiyatlarga ega bo’lishi kerak:

1. U odatda shartli ravishda optimal bo’lmasa ham, yahshi yechimlarni topish kerak.
2. Uni ixtiyoriy ma’lum aniq algoritimdan ko’ra, amalga oshirish tezroq va soddaroq bo’lishi kerak.

Odatda yahshi algoritmlar evristik bo’lib chiqadi. Faraz qilaylik, biz tez ishlaydigan va barcha test topshiriqlariga javob beradigan algoritimni tuzdik, lekin uning to’g’riligini isbotlab bilmaymiz. Shunday isbot berilmaguncha, algoritim evristik deb tushuniladi.

Misol tariqasida quyidagi algoritimni ko’rib chiqamiz:

**GTS algoritmi:** (kommivoyajer). N ta shaharlar va C narxlar matrisasi berilgan kommivoyajer masalasi uchun U shahardan boshlab COST narxli TOUR yaqinlashgan yechimni topish kerak.

**Qadam 0:** [Insiallashtirish] TOUR:=0; COST:=0; V:=U;

**Qadam 1:** [Hama shaharlarni o’tish] For k:=1 to N-1 do qadam 2;

**Qadam 2:** [Keyingi vektorga o’tish]

Faraz qilaylik, (V,W) – V shahardan W ga olib borayotgan eng kichik narxli vektor. Unda:

TOUR:=TOUR+(V,W); COST:=COST+C(V,W);

V:=W;

**Qadam 3:** [Marshrutni tugatish] TOUR:=TOUR+(V,1);

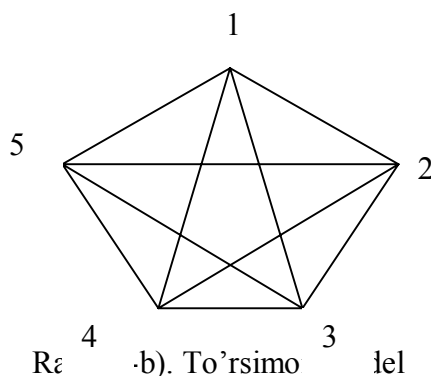
COST:=COST+C(V,1);

Marshrutni tasvirlash uchun biz matematikada graf yoki tur deb nomlanayotgan chizmadan foydalanamiz. Umuman tur – bu nuqtalar va bir nechta yoki barcha ikki nuqtalarni bog’layotgan chiziqlar to’plami, undan tashqari chiziqlar ustida qiymatlar ham berilishi mumkin.

Masalani soddallashtirish uchun beshta shahar uchun yechim topamiz. Rasm. 1a – narxlar matrisasi. Rasm. 1b – turli model ko’rsatilgan.

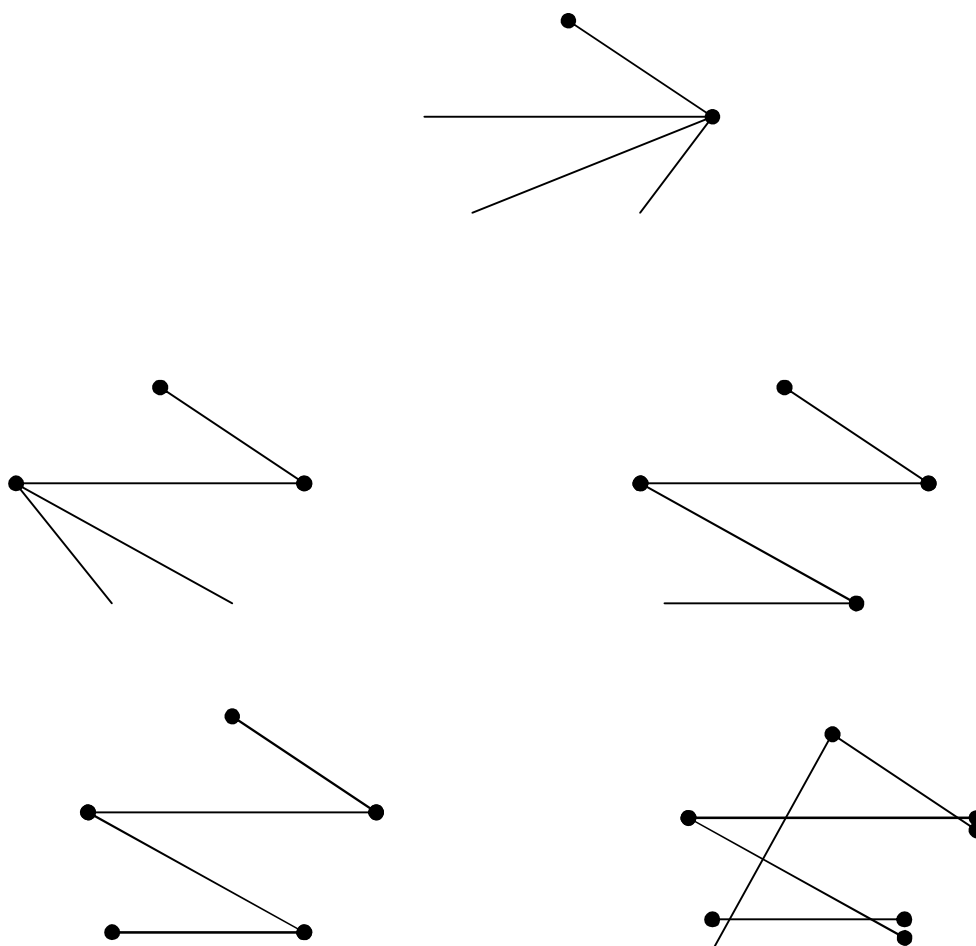
-	1	2	7	5
1	-	4	4	3
2	4	-	1	2
7	4	1	-	3
5	3	2	3	-

Rasm 1-a). Narxlar matrisasi



Turlar nazariyasida shaharlar ruyhati bir shahardan boshlab va o’sha shaharga barcha qolgan shaharlarni bir martadan o’tib qaytib kelish jarayonini belgilaydi. Bunday o’tishni marshrut deb ta’riflaymiz. Marshrut narxi chiziqlar ustidagi qiymatlar yig’indisi bilan aniqlanadi.

Rasm 2 algoritim GTS bo’yicha K marshrutni shahar1 dan boshlab tuzishni aks ettiradi.



Rasm 2. Algoritm qadamlari

GTS algoritmi bo'yicha marshrut narxi 14 ga teng. Bu yerda algoritm eng kichik narxli marshrutni topmaganini ko'ramiz. Masalan, marshrut 1-5-3-4-2-1 narxi  $5+2+1+4+1=13$ .

Odatda yaqinlashgan algoritmlar tez bo'lsa ham, hamma vaqt optimal yechimni berolmaydilar. GTS algoritmi uchun biz nazoratchi nisol topib bildik. Lekin, yaqinlashgan algoritmi ishlamasligini isbotlash hamma vaqt ham oson bo'lmaydi.

GTS algoritmi uchun dastur yozish ancha yengil. Lekin uni tezligini tahlil qilib ko'raylik. Ixtiyoriy kommivoyajer masalasi uchun (besh shahardan iborat)  $C$  narxlar matrisasini o'qish va tuzishga  $O(N^2)$  operatsiya kerak. Demak, pastki murakkablik chegara algoritmi uchun  $O(N^2)$  teng va GTS algoritmini yahshi evristik algoritmi deb qabul qilishimiz mumkin.

#### Takrorlash ucun savollar

1. Masala qo'yilishidagi o'zgaruvchilarni aniqlang.
2. Evristik algoritmlarni ta'riflab bering.
3. GTS algoritmini tuzishdagi qadamlarni aytib bering.
4. Algoritmni baholash jarayonini tavsiflab bering.

## 11 - MAVZU: SHOHLAR VA CHEGARALAR USLUBI

### Reja

1. Masala qo'yilishi.
2. To'rsimon modellardan foydalanish.
3. Shoxlar bo'yicha baholash.
4. Chegaralar bo'yicha baholash.

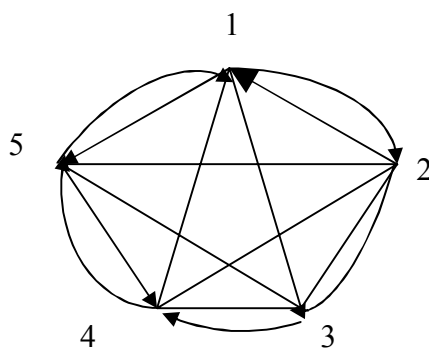
Bu usul yechimlar fazosining tursimon modelini ta'qiq qiladigan usullar turiga kiradi va kombinatorika masalalarining keng doirasiga qo'llanilishi mumkin.

Bunday algoritmlar ko'proq optimizatsiyaga yo'naltirilgan va ancha murakkab bo'ladi, lekin kommivoyajer masalasini yechishda juda qulay hisoblanadi.

Masalani tarmoqlanish ko'rinishida tadqiq qilamiz. Quyidagi rasmlarda beshta shahar uchun kommivoyajer assimetrik masalasining narxlar matrisasi berilgan.

	1	2	3	4	5
1	-	25	40	31	27
2	5	-	17	30	25
3	19	15	-	6	1
4	9	50	24	-	6
5	22	8	7	10	-

Rasm 1. Narxlar matrisasi



Rasm 2. To'rsimon model

Bundan tashqari rasmda narxlarni ko'rsatish uchun yo'naltirilgan tarmoqdan foydalanamiz. Bu yerda  $i$  shahardan  $j$  shaharga borish bahosi,  $j$  dan  $i$  ga borish bahosiga teng bo'lishi shart emas. Bizning izlash daraxtimizning ildizi barcha mumkin bo'lgan marshrutlar to'plamiga mos bo'ladi, ya'ni besh shahar masalasidagi  $(4!)$  marshrutlar to'plamini aks ettiradi. Umuman, ixtiyoriy  $N$  shaharni assimetrik masala uchun ildiz barcha  $\{(N-1)!\}$  mumkin bo'lgan marshrutlar  $R$  to'plamini akslantiradi. Ildizdan tarqaladigan shohlar bir qirrani, masalan,  $(i,j)$  – ni tanlash bilan aniqlanadi. Bu ishdan maqsad – barcha marshrutlar to'plamini ikki to'plamga ajratish: Biri optimallashtirilgan tur, ikkinchisi esa optimallashtirilmagan turlardan iborat bo'ladi.  $(i,j)$  tanlangan qirra optimal turga tegishli deb hisoblagan holda,  $R$  to'plamni ikkiga bo'lamiz, ya'ni  $\{i,j\}$  va  $\{\bar{i},\bar{j}\}$  to'plamlarga.  $\{i,j\}$  to'plamiga  $(i,j)$  qirrasini qatnashgan turlar kiradi,  $\{\bar{i},\bar{j}\}$  to'plamga esa shu qirra qatnashmagan tur.

Faraz qilaylik, biz tarmoqlanishni  $\{i,j\}=\{3,5\}$  qirrasida amalga oshirdik, chunki bu qirraning bahosi matrisada eng kichik. Unda rasmda ildiz va uning birinchi darajasini ko'rsatishimiz mumkin.

Shuni ta'kidlash kerakki, R-ga tegishli har bir tur birinchi darajaning faqatgina bitta to'plamiga kiradi. Agar biz  $\{3,5\}$  to'plamida optimal tur yo'qligini qabul qilsak,  $\{3,5\}$  to'plamini tadqiq qilishga o'tamiz.  $\{3,5\}$  to'plamini ham yuqoridagidek bo'lamiz. Arzonlik bo'yicha (2,1) qirrasida ikkinchi o'rinda  $C(2,1)=5$ . Shuning uchun  $\{3,5\}$  to'plamini  $Y$  va  $\underline{Y}$  deb belgilaymiz.  $Y$  to'plamga  $X$  to'plamda qatnashgan va  $(i,j)$  qirrasida mavjud turlar kiradi,  $\underline{Y}$  to'plamga  $(i,j)$  qirrasida qatnashmagan  $X$  ning qism to'plami.

Yuqorida tadqiq qilingan jarayon tarmoqlanish haqida tasavvur beradi. Endi chegaralar hisoblashni ko'ramiz.

Har bir daraxt uchi bilan shu uch bilan belgilangan to'plamning ixtiyoriy turining pastki narx chegarasini bog'laymiz. Bunday chegaralarni hisoblash – shoxlar va chegaralar kabi usullarda tadqiqotlarni yengillashtirish uchun asosiy faktordir. Shuning uchun ularni aniq hisoblashga katta e'tibor berish lozim.

Sababi quyidagicha: Masalan,  $m$  baholi konkret bir turni qabul qilaylik. Unda, agar  $V_k$  uchi bilan belgilangan turlar to'plami bilan bog'liq pastki chegara  $M \geq m$  bo'lsa, optimal turni izlash jarayoni davomida  $V_k$  va undan keyingi uchlarni tadqiq qilish kerak bo'lmay qoladi.

Xulosa qilib, shuni aytish mumkin-ki, shoxlar va chegaralar uslubi murakkab bo'lsa-da, kommivoyajer masalasi katta sonli shaharlar va narxlar bilan berilganda, algoritmlar aniq va tez ishlaydi, algoritmlarning murakkabligi esa eksponensial.

#### Takrorlash uchun savollar

1. *Kommivoyajer masalasida ikki tomonli narxlar matrisasi qaysi holatda tuziladi.*
2. *To'rsimon modellardan foydalanish.*
3. *Shoxlar bo'yicha baholashning afzalligini tushuntirib bering.*
4. *Chegaralar bo'yicha baholash nimadan iborat?*

## 12 - MAVZU: ENG QISQA YO'LLAR. DEYKSTRA ALGORITMI.

### *Reja*

1. *Eng qisqa yo'llar masalalarining turlari.*
1. *Sozli algoritmi tuzish.*
3. *Algoritmi psevdokodda ishlab chiqish*
4. *Algoritmi baholash.*

Yo'l tarmoqlari atlas (karta) qismi berilgan bo'lib, undan A va B nuqtalar orasidagi "eng yahshi" marshrutni topish kerak bo'lsin. "Eng yahshi" marshrutni ko'p faktorlar belgilashi mumkin, masalan, tezlik cheklangan holda marshrutni o'tish vaqti, o'tish kerak bo'lgan shaharlar soni va boshqalar.

Biz masalani eng qisqa yo'llar faktori bo'yicha yechamiz. Masalaning modeli turlar yordamida tuziladi. Uzlüksiz G turni har bir qirrasiga uning uzunligiga teng qiymat berilgan ko'rinishida tuzamiz. Bunday turda masofa irrallar yig'indisiga teng bo'ladi. Masalaning maqsadi ikkita berilgan uchlarni orasidagi eng qisqa marshrutni topishdir.

Umuman, eng qisqa yo'llar masalalari kombinator optimallashtirishning fundamental muammolaridandir. Ularning bir necha turlari mavjud, masalan, ikkita berilgan uchlar orasida, berilgan va qolgan barcha uchlar orasida, turdagi har bir uchlar juftliklari orasida va boshqalar.

### **Deykstra algoritmining so'zli tavsifi**

Shunday masalalarni yechish uchun Deykstra algoritmi ancha qulay va yahshi deb topilgan.

Algoritm quyidagi qadamlardan iborat:

1. Dastlab, berilgan (**Lex**) uchidan qolgan barcha uchlargacha bir qirra uzunligidagi masofalar aniqlanadi.
2. Ulardan eng qisqasi "doimiy eng qisqa masofa" sifatida belgilanadi (**Lex** va **BVa** uchlari qirrasini).
3. Aniqlangan masofa **BVa** dan boshqa bor uchlargacha masofalarga qo'shiladi.
4. Hosil bo'lgan yig'indilar dastlab aniqlangan **Lex** dan qolgan uchlargacha bo'lgan masofalar bilan taqqoslanadi. Natijada masofasi qisqaroq bo'lgan uchning qirrasini tanlanadi.
5. **BVa** uchi, eng qisqa masofa aniqlangan uch sifatida, ruyhatdan o'chiriladi. Ruyhatga boshqa uch qo'yiladi, masalan, **Roa**. **Bva** o'z navbatida, boshqa, izlanayotgan ruyhatga qo'yiladi.

Keyingi eng qisqa masofani topish uchun butun jarayon qayta bajariladi. **BVa** dan keyin yana bir uch ruyhatga qo'yiladi. Dastlabkisi esa ruyhatdan o'chiriladi. Sikl **Bed** va **Lex** uchlarini bog'lash uchun belgilangan qirralar aniqlanishi bilan to'xtatiladi.

Rasm bo'yicha ikkinchi iteratsiyada **Nbr** uchi aniqlanadi va **Roa** gacha masofa 41 deb qabul qilinadi. Uchinchi iteratsiyada **Gla** uchigacha masofa eng qisqa va 27 deb qabul qilinadi. Quyidagi rasmda eng qisqa yo'llar daraxti ko'rinishida ularning natijaviy to'plami keltirilgan.

Aylanalar ichidagi sonlar algoritm bo'yicha qirralar tanlanish tartibini ko'rsatadilar. Bu daraxt bo'yicha biz **Lex** uchidan ixtiyoriy bizni qiziqtirayotgan uchgacha eng qisqa yo'lni topishimiz mumkin.

Ko'rilgan misolda **Bed** uchi **Lex** dan boshlab eng oxirgi bo'lib chiqdi, ya'ni **Lex** dan **Bed** gacha eng qisqa masofani topish uchun biz **Lex** dan barcha qolgan uchlargacha eng qisqa yo'llarni topishga majbur bo'ldik.

Demak, eng yomon holatda 2 ta berilgan uchlar orasidagi eng qisqa yo'lni topish, bir berilgan nuqtadan qolgan barcha nuqtalargacha eng qisqa yo'l topish masalasi bilan murakkabligi bir xil bo'ladi.

### **Algoritmni psevdokodda ishlab chiqish**

#### **1. Masala qo'yilishi.**

$M$  ta uch va  $N$  ta qirralardan iborat uzluksiz grafda  $V_0$  uchidan  $W$  uchigacha  $Dist(W)$  masofani topish kerak. Qirralar uzunliklari  $A$  matrisa bilan berilgan deb hisoblaymiz.

**Qadam 0.** [uchlarni belgilash] – bu yerda  $V_0$  uchini "aniqlangan" deb belgilaymiz, qolgan barcha uchlarini "aniqlanmagan" deb hisoblaymiz.

**Qadam 1.** [o'zgaruvchilarni inetsiallashtirish] – bu yerda

$Dist(U) := A(V_0, U)$  – barcha  $G$  ga tegishli  $U$  uchlari uchun;

$Dist(V_0) := 0$ ;  $Next := V_0$ ;

**Qadam 2.** [sikl]. *While*  $NEXT \neq W$  *do*

*Begin*

**Qadam 3.** ["aniqlanmagan" uchgacha eng qisqa yo'lni yangilash]. Har bir "aniqlanmagan"  $U$  uchi uchun

$Dist(U) := \min(Dist(U), Dist(Next) + A(Next, U))$ .

**Qadam 4.** ["aniqlanmagan" uchgacha eng qisqa yo'lni tanlash]. Agar  $U$  barcha "aniqlanmagan" uchlari orasida  $Dist(U)$  masofasi eng kichik bo'lsa, uni "aniqlangan" deb belgilaymiz va  $NEXT := U$ .

**end.**

Bu algoritmning va dasturning murakkabligini  $O(M^2)$  ekanligini ko'rsatish mumkin.

Takrorlash ucun savollar

1. Qaysi mezonlar bo'yicha eng qisqa yo'llar masalalarini yechish mumkin?
2. Deykstra algoritmi nima uchun yaxshi hisoblanadi?
3. Algoritmni psevdokodda tavsiflashning qo'layligini ko'rsating

#### 4. Deykstra algoritmining bahosi qanday?

### 13 - MAVZU: TARTIBLASH ALGORITMLARI. XOARA USULI.

#### *Reja*

1. *Tartiblash masalalarining turlari*
2. *Xoaraning tartiblash algoritmi mazmuni*
3. *Xoara algoritmini rekursiv usulda amalgam oshirish*
4. *Algoritmi baholash*

#### **Tartiblash masalalarining turlari**

Umuman tartiblash deganda berilgan ob'yektlar to'plamini ma'lum tartibda joylash uchun qayta ishlash jarayoni tushuniladi.

Tartiblash natijasida to'plamdagi elementlarni izlash jarayonlari yengillashadi. Undan tashqari tartiblashlar misolida qanday qilib algoritmi murakkablash evaziga samaradorlikni oshirishga erishish mumkinligini ko'rsatsa bo'ladi.

Hozirgi kunda ko'pgina tartiblash algoritmlari mavjud. Algoritmi tanlash qayta ishlanayotgan ma'lumotlar strukturasi bog'liq va shu sababli tartiblash usullari asosan 2 sinfga ajratiladi. Bular massivlarni tartiblash va fayllarni tartiblash. Ularni yana ichki va tashqi tartiblash ham deb nomlaydilar. Chunki massivlar mashinaning tezkor xotirasida joylashadi. Fayllar esa odatda ancha hajmi katta bo'lgan lekin sekin ishlaydigan tashqi xotiradan olinadilar.

#### **Xoaraning tartiblash algoritmi mazmuni**

Eng yahshi tartiblash algoritmlaridan biri deb Ch. Xoara usuli hisoblanadi. Bu usul almashuvga asoslangan.

Bu yerda yahshi samaradorlikka erishish uchun dastlab katta masofadagi ya'ni bir-biriga eng uzoq joylashgan elementlarni almashtirish qo'llaniladi. Faraz qilaylik bizda  $n$  ta element kalitlar bo'yicha qayta tartibda berilgan. Xoara usuli bo'yicha ularni  $\frac{n}{2}$  ta almashuv bilan tartiblash mumkin. Buning uchun dastlab eng chap va eng o'ng tomonda joylashgan elementlarni almashtiramiz. Keyin ikki tomondan o'rtaga qarab kelamiz. Lekin bu faqatgina qayta tartib aniq bo'lganda amalga oshiriladi.

Endi massiv ixtiyoriy tartibda berilgan bo'lsin. Ixtiyoriy  $X$  elementni tanlab massivni chapdan o'ngga qandaydir  $a_i > x$  element uchramaguncha ko'rib chiqamiz. Keyin massivni o'ngdan chapga qandaydir  $a_j < x$  element uchramaguncha o'tamiz.

$a_i$  va  $a_j$  elementlarni o'rinlarini almashtirib massivni ikki tomondan ko'rib chiqish jarayonini massiv o'rtasiga kelmaguncha davom ettiramiz. Natijada massiv 2 qismga bo'linadi. Chap qismdagi elementlar  $x$  dan katta yoki teng bo'ladilar. O'ng tomondagi elementlar  $x$  dan kichik yoki teng bo'ladi.

Dastur tuzayotganda bu jarayonni prosedura yordamida amalga oshirish mumkin. Prosedurani rekursiv va norekursiv usullar bilan tuzish mumkin.

### Xoara algoritmini rekursiv usulda amalga oshirish

Quyidagi dastur rekursiv prosedurani qo'llaydi.

```
Procedure Hoare;
  Procedure sort (L, R: index);
    var i, j: index; w, x: item;
    begin i:=L; j:=R; x:=a[(L+R) div 2];
    repeat
      while a[i]<x do i:=i+1 end;
      while a[j]>x do j:=j-1 end;
      if i<=j then
        begin w:=a[i]; a[i]:=a[j]; a[j]:=w;
          i:=i+1; j:=j-1; end;
    until i>j
    if L<j then sort (L, j) end;
    if i<R then sort (i, R) end;
  end {*sort*};
begin sort (1, n);
end {* Hoare*};
```

Norekursiv dasturni tuzish uchun yordamchi steklardan foydalaniladi.

### Algoritmni baholash

Xoara algoritmi unumdorligini tahlil qilamiz. Boshlab bo'linish jarayonini ko'raylik. Qandaydir  $x$  ni tanlab biz massivni to'liq o'tamiz. Demak,  $n$  ta taqqoslashni amalga oshiramiz. Taqqoslashlarni umumiy soni  $n \cdot \log(n)$  ekanligi, o'rin almashtirishlar soni esa  $\frac{n \cdot \log(n)}{6}$  ekanligi isbotlangan.

Bizning misolimizda  $x$  - o'rtacha element deb tanlangan, lekin Xoara fikri bo'yicha  $X$  ixtiyoriy tanlanishi kerak. Xoara algoritmining o'rtacha ishlash vaqti  $O(n \cdot \ln(n))$  teng.

### Takrorlash ucun savollar

1. Tartiblash masalalarining qaysi turlarini bilasiz?
2. Xoaraning tartiblash algoritmi mazmuni nimada?
3. Xoara algoritmini qanday usullar bilan amalga oshirish mumkin?
4. Xoara algoritmi bahosini tavsiflab bering.



1. *Masala quyilishi.*
2. *Matrisalarni ko'paytirish uchun lemma.*
3. *Matrisalarni ko'paytirish uchun teorema.*
4. *Algoritmni baholash.*

Matrisalarni ko'paytirishda assimpotik hisoblash murakkabligini tahlil qilamiz. Oddiy  $O(n^3)$  murakkabligiga ega algoritmni assimpotik yaxshilash jarayonini o'rganamiz va 2 ta  $n \times n$  matrisalarni ko'paytirishda  $O(n^{2,81})$  vaqt yetarli ekanligini ko'rsatamiz.

Ikkita  $n \times n$  o'lchamli **A** va **B** matrisalar berilgan bo'lsin. Bu yerda  $n - 2$  ning darajasidagi son deb qabul qilinadi. **A** va **B** matrisalarni 4 ta  $\frac{n}{2} \times \frac{n}{2}$  o'lchamli matrisalarga bo'lish mumkinligi haqidagi lemmani qo'llab, **A** va **B** larni ko'paytmasini quyidagicha tasvirlaymiz.

$$\begin{vmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{vmatrix} \cdot \begin{vmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{vmatrix} = \begin{vmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{vmatrix}$$

$$C_{11} = A_{11}B_{11} + A_{12}B_{21}, \quad C_{12} = A_{11}B_{12} + A_{12}B_{22}, \quad C_{21} = A_{21}B_{11} + A_{22}B_{21}, \quad C_{22} = A_{21}B_{12} + A_{22}B_{22}$$

$$C_{ik} = \sum_{j=1}^n a_{ij} \cdot b_{jk}$$

Bu yerda

$A_{11}$  – yuqori chap kvadrat,

$A_{12}$  – yuqori o'ng kvadrat,

$A_{21}$  – pastki chap kvadrat,

$A_{22}$  – pastki o'ng kvadrat.

Shtlassen 2 ta  $2 \times 2$  o'lchamli matrisalarni ko'paytirish uchun sun'iy usulni ishlab chiqqan. Bu usulda 7 ta ko'paytirish amalga oshiriladi. Usulni rekursiv qo'llab, 2 ta  $n \times n$  matrisani  $O(n^{\log 7})$  vaqtda ko'paytiriladi.  $O(n^{\log 7}) \approx n^{2,81}$ .

**Lemma.** 2 ta  $2 \times 2$  o'lchamli matrisalarni ko'paytirishda 18 ta qo'shish va ayirish va 7 ta ko'paytirish amallari bajarilsa yetarli.

**Isbot.**  $C = a * b$  ko'paytmasini hisoblashdan oldin quyidagi ko'paytirishlarni amalga oshiramiz.

$$m_1 = (a_{12} - a_{22})(b_{21} + b_{22});$$

$$m_2 = (a_{11} + a_{22})(b_{11} + b_{22});$$

$$m_3 = (a_{11} - a_{21})(b_{11} + b_{12});$$

$$m_4 = (a_{11} + a_{12}) \cdot b_{22};$$

$$m_5 = a_{11} \cdot (b_{12} - b_{22});$$

$$m_6 = a_{22} \cdot (b_{21} - b_{11});$$

$$m_7 = (a_{21} + a_{22}) \cdot b_{11};$$

Keyin  $c(i,j)$  larni quyidagi formulalar bo'yicha hisoblaymiz:

$$c_{11} = m_1 + m_2 - m_4 + m_6; \quad c_{12} = m_4 + m_5; \quad c_{21} = m_6 + m_7; \quad c_{22} = m_2 - m_3 + m_5 - m_7;$$

Amallar soni oddiy hisoblanadi.

**Teorema.** Ikkita  $n \times n$  o'lchamli matrisalarni  $O(n^{\log 7})$  sonli arifmetik amallarni bajarib ko'paytirish mumkin.

**Isbot.** Birinchi,  $n - 2$ -ning darajasidagi son bo'lgan hoholatni ko'ramiz. Ikki  $n \times n$  o'lchamli matrisalarni ko'paytirish uchun kerak bo'lgan arifmetik operatsiyalar sonini  $T(n)$  deb belgilaymiz. Unda

lemma bo'yicha  $T(n) = 7T\left(\frac{n}{2}\right) + 18\left(\frac{n}{2}\right)^2$ ,  $n \geq 2$  uchun.

$n$  o'lchamli masala qandaydir chiziqli vaqtda ikkita  $\left(\frac{n}{2}\right)$ -o'lchamli masalalarga bo'linishi  $O(n \log n)$

murakkablikdagi algoritmnin beradi. Shuni hisobga olib  $T(n) = O(n^{\log 7})$  ekanligini ko'rsatamiz.

Endi  $n - 2$ -ning darajasidagi son bo'lmagan holatni ko'ramiz. Bu holda har bir matrisani shunday matrisaga qo'yamiz-ki, uning tartibi  $n$  dan katta bo'lgan  $2$ -ning darajasidagi eng kichik son bo'lsin. Berilgan matrisamizning tartibi bu holda  $2$  baravarga ham oshmaydi, bu esa  $T(n) = O(n^{\log 7})$  hamma  $n \geq 1$  lar uchun chin ekanligini ko'rsatadi.

#### Takrorlash ucun savollar

1. Matrisalarni ko'paytirish oddiy usulda qanday amalgam oshiriladi?
2. Matrisalarni ko'paytirish uchun lemmani aytib bering.
3. Matrisalarni ko'paytirish uchun teoremani aytib bering.
4. Xoara algoritmi bahosini tavsiflab bering.

## 15 - MAVZU: TYURING MASHINASI

### Reja

1. Tyuring mashinasining yaratilish tarixi.
2. Tyuring mashinasi algoritmi ishlash prinsipi.
3. Tyuring mashinasi algoritmi imkoniyatlari

Tyuring mashinasi – bu 1936 yilda ingliz olimi A. Tyuring tomonidan universal algoritmik model sifatida taklif qilingan abstrakt mashina. U uch qismdan iborat: lenta, golovka va boshqaruv qurilma (rasm 1).

Lenta ikki tomonga cheksiz uzun va katakchalarga bo'lingan. Har bir katakchada faqat bitta belgi yoziladi. Mumkin bo'lgan simvollar soni chekliva mashinaning  $A = \{a_1, \dots, a_n\}$  deb belgilangan alfavitni tashkil qiladi. Simvol yo'qligi maxsus bo'sliq belgisi bilan ko'rsatiladi.

Golovka hamma vaqt lentaning biror-bir katakchasi ustida joylashgan bo'ladi. U katakchalarni o'qiydi, yozadi, o'chiradi va lenta bo'ylab yurishi mumkin. Golovka har bir ish siklida  $Q\{q_1, q_2, \dots, q_n\}$  chekli to'plamga tegishli holatlardan birida bo'ladi. Holatlar ichidan  $q_1$  - boshlang'ich va  $q_n$  -yakuniy holatlarni ajratib ko'rsatishi mumkin.

Tyuring mashinasining bir elementar qadami quyidagilardan iborat. Golovka qaysi yacheyka ustida joylashgan bo'lsa, shunga yozilgan simvolni o'qiydi. Simvol qiymati va o'zining holatidan kelib chiqib,

golovka yangi holatga o'tadi va qaysi simvolni yozish va qanaqa harakatni bajarish ko'rsatilgan komandani amalga oshiradi. Shunday qilib, mashina keyingi qadamni bajarishga tayyor.

Mashina ishlaydigan qoidani quyidagicha ko'rsatish mumkin:  $q_i a_j \rightarrow q'_i a'_j d_k$ . Bu yerda  $q_i, q'_i$ -har xil holatlar,  $a_j, a'_j$ -o'qiladigan va yoziladigan simvollar,  $d_k$ -golovkaning harakati. Bu hharakat uch xil bo'lishi mumkin: bir katakcha chapga, bir katak o'ngga, joyida qolish. Har bir  $q_i a_j$  kombinatsiyasi uchun faqat bitta qoida ishlaydi. Lekin  $q_n$  uchun qoida yo'q, chunki mashina to'xtaydi.

Konkret Tyuring mashinasi  $A$  va  $Q$  elementlarini hamda qoidalar to'plamini belgilash bilan beriladi.  $A, Q$  to'plamlar va qoidalarhillari cheksiz ko'p bo'lishi mumkin, shu sababli konkret Tyuring mashinalari ham cheksiz ko'p bo'lishlari mumkin.

Qoidalar to'plami odatda jadval ko'rinishida beriladi. Satrlar bo'yicha holatlar, ustunlarga simvollar qo'yiladi.  $q_i$  satri va  $a_j$  ustuni kesishmasida esa uchta belgi  $q_i a_j d_k$  qo'yiladi. Har bir bo'sh bo'lmagan jadval katakchasiga qandaydir qoida mos keladi, bo'sh katakcha esa qoida yo'qligi va keraksizligini ko'rsatadi, chunki bu  $q_i$  holatda golovka hech qachon  $a_j$  simvolga uchramaydi.

Tyuring mashinasi yordamida xilma-xil turdagi funksiyalarni hisoblash mumkin: sonli, mantiqiy, simvulli. Funksiyani turi, odatdagidek, kirish va chiqish ma'lumotlari bilan belgilanadi. Umuman Tyuring tezligi bo'yicha ixtiyoriy isoblanadigan funksiyaga (agar uni hisoblaydigan algoritm mavjud bo'lsa) Tyuring mashinasini ko'rish va qo'llash mumkin.

### Adabiyotlar ro'yhati

1. Жуманов И.И., Кобилов С.С. СУБД и информационные системы. Уч. пособие. Самарканд, 1977 г.
2. В.А.Успенский, А.Л.Семенов. Теория алгоритмов: основные открытия и приложения. – М: Наука, 1987, 287 с.
3. Т.Кормен, Ч.Лейзерсон, Р.Ривест. Алгоритмы: построение и анализ. Сер: Классические учебники. М.: МЦНМО, 2001.- 960 с.
4. Гуломов С.С. ва бошқалар. Ахборот тизимлари ва технологиялари. Тошкент, 2000 й.
5. Д.Кнут. Искусство программирования для ЭВМ. Основные алгоритмы.-М: Мир, 2000 г.
6. Уотермен Д. Руководство по экспертным системам. М: Мир, 1989 г.
7. А.Ахо., Дж.Хопкрофт. Построение и анализ вычислительных алгоритмов. - М: Мир, 1979 г., 535 с.
8. Лебедев В.И. Введение в системы программирования. М: Статистика, 1975 г.
9. Донован Дж. Системное программирование. М: Мир, 1975г.
10. Интеллектуализация ЭВМ. Перспективы развития вычислительной техники. Под ред. Ю.М.Смирнова. М: 1989 г.
11. Тыугу Х. Концептуальное программирование. М: Наука, 1984.
12. Попов В.В. Общение с ЭВМ на естественном языке. М:Наука, 1982.
13. Построение экспертных систем. Пер. с англ. Под ред. Хенес-Рота Р., А.Уотермана, А.Лента. М: Мир, 1987 г.
14. Н. Вирт. Алгоритмы и структуры данных. – Досса, Хамарайан, 1997.
15. Шафрин Ю. Основы компьютерной технологии: Справочник школьника. М, 1997.

16. Таусенд К., Фохт Д. Проектирование и программная реализация экспертных систем на персональных ЭВМ. – М: Финансы и статистика, 1990.
17. Нортон Н. Программно-аппаратная организация IBM PC. – М: Мир. 1991.

#### Qo'shimcha adabiyotlar

1. Жуманов И.И., Мингбоев Н.С. Ҳисоблаш системаларининг информацион асослари. Самарқанд,: СамДУ нашри, 2002, 107 бет.
2. Мингбаев Н.С., Жуманов И.И. Информатика.- Самарқанд,: СамДУ нашри, 2002, 107 бет.
3. Мингбаев Н.С., Жуманов И.И. Компьютер технологиялари- Самарқанд,: СамДУ нашри, 2004, 152 бет.
4. Жуманов И.И., Мингбоев Н.С. Ахборот технологиялари (1-қисм: ахборот технологияларининг қурилмавий ва дастурий таъминоти), Самарқанд,: СамДУ нашри, 2005, 148 бет.
5. Жуманов И.И., Мингбоев Н.С. Ахборот технологиялари (2-қисм: Ахборот технологияларининг информацион таъминоти), Самарқанд,: СамДУ нашри, 2005, 70 бет.

#### 5. AMALIY MASHGULOTLARGA DOIR USLUBIY KO'RSATMALAR

О'zbekiston Respublikasi  
Oliy va o'rta maxsus ta'lim vazirligi

Alisher Navoiy nomidagi  
Samarqand Davlat universiteti

Axatov A.R., Abdullayev A.N.

**Algoritmlash va dasturlash**

**(uslubiy qo'llanma)**

## **SAMARQAND – 2010**

Algoritmash va dasturlash: uslubiy qo'llanma. – Samarqand: SamDU nashri, 2007. – 69 bet.

Ushbu uslubiy qo'llanma amaliy matematika va informatika bakalavriat ta'lim yo'nalishidagi "Algoritmsh nazariyasi", "Kompyuter amaliyoti", "Dasturlash asoslari" fanlarini o'rganuvchi 1 va 2-bosqich talabalariga mo'ljallangan.

Uslubiy qo'llanma 6 bo'limdan iborat bo'lib, har bir bo'limda qisqa nazariy ma'lumotlar keltirilgan, namuna sifatida echib ko'rsatilgan bir nechta masalalar uchun sinov qiymatlari, algoritm, blok-sxema, Turbo Paskal tilidagi dastur matni berilgan. Mavzularni mustahkamlash maqsadida mustaqil ishlash uchun topshiriqlar bilan to'ldirilgan.

Qo'llanma zamonaviy kompyuter va dasturlash texnologiyalarini mustaqil ravishda o'rganayotgan barcha qiziquvchilar uchun ham foydali.

Tuzuvchilar: Axatov A.R., Abdullayev A.N.

Mas'ul muharrir: texnika fanlari doktori,  
professor Jumanov I.I.

Taqrizchilar: fizika-matematika fanlari nomzodi, dotsent  
Bekmurodov Q.A.,  
texnika fanlari nomzodi, dotsent To'raqulov I.N.

©Alisher Navoiy nomidagi Samarqand Davlat universiteti, 2007

## SUZ BOSHI

Dasturlarni mustaqil tuzishdan maqsad kompyuterga mutloq xokimlik qilish, ya'ni ish davomida yuzaga keladigan muammolarni tezroq hal etish imkonini yaratishdir. Kompyuter dasturlari sermehnat ishlarni avtomatlashtiradi, xatolarni kamaytiradi va mehnat unumdorligini oshiradi. Bundan tashqari, dasturlar tuzish juda ham mashg'ulotdir.

Dasturlarni yaratish jarayonida qo'yilgan masalaning yechish algoritmi dastlab to'g'ri ishlab chiqilishi muhim ahamiyatga ega. Shuning uchun algoritmlarni tuzish va dasturlarni ishlab chiqish bir-biri bilan chambarchas bog'liq jarayonlardir. Oliy o'quv yurtlarining informatika, axborot texnologiyalari, amaliy matematika kabi yo'nalishlarida ta'lim olayotgan talabalar algoritmlarni ishlab chiqish, dasturlar yaratish, ularni sinash, sozlash, tahlil qilish uchun bilimlarni puxta o'zlashtirishlari zarur. Bunda, ta'lim oluvchi uchun dasturlarni ishlab chiqishda asosiy va eng muhim bosqich hisoblangan algoritmlarni tuzish va shular asosida dasturlar yaratish haqida ma'lumotlarni beruvchi adabiyotlar kerak.

Qo'llanma chiziqli, tarmoqlanuvchi, takrorlanuvchi, murakkab jarayonlarining algoritmlarini ishlab chiqish qoidalarini o'rgatishga mo'ljallangan. Uslubiy qo'llanmada har bir keltirilgan mavzuga doir qisqa nazariy ma'lumot, namuna sifatida yechib ko'rsatilgan bir nechta masala uchun sinov qiymatlari, algoritmlar, blok-sxema, Turbo Pascal tilida dastur matni berilgan. Mavzuni mustahkamlash maqsadida mustaqil ishlash uchun topshiriqlar bilan to'ldirilgan.

Qo'llanma Oliy o'quv yurtlari talabalari uchun mo'ljallab yozilgan va zamonaviy kompyuter texnologiyalarini mustaqil ravishda o'rganayotgan barcha qiziquvchilar uchun ham foydalidir.



## MUNDARIJA

1. Chiziqli va tarmoqlanuvchi jarayonlarni dasturlash . . . . .	5
2. For (uchun) tipidagi sikl yordamida bajariladigan algoritm va dasturlar tuzish . . . . .	11
3. For (uchun) tipidagi ichma-ich joylashgan sikl yordamida bajariladigan algoritm va dasturlar tuzish. . . . .	19
4. While (toki) tipli sikl yordamida bajariladigan algoritm va dasturlar tuzish. . . . .	30
5. Toki tipidagi ichma-ich joylashgan sikl yordamida bajariladigan algoritm va dasturlar tuzish. . . . .	43
6. For (uchun) va While (toki) tipli sikllar kombinatsiyasi yordamida algoritm va dasturlar tuzish. . . . .	57
ADABIYOTLAR . . . . .	68

### ***1. Chiziqli va tarmoqlanuvchi jarayonlarni dasturlash***

**1.1 - misol.** *Asosining radiusi  $R$ , balandligi  $H$  bo'lgan silindrni yon sirtining yuzasi va hajmini aniqlang.*



### Test

Berilganlar		Natija	
R = 1	H = 1	V = 3.14	S = 6.28

*Algoritmi:*

**alg** Silindr (haq R, H, V, S)

**arg** R, H

**natija** V, S

**boshl** haq Pi

Pi := 3.14

V := Pi \* R\*\*2 \* H

S := 2 \* Pi \* R \* H

**tamom**

*Turbo Pascaldagi dasturi:*

Program Silindr;

Var

R, {silindr asosi radiusi}

H, {silindrning balandligi }

V, {silindr hajmi }

S: Real; {silindr yon sirti yuzasi}

BEGIN

Write(' Silindrning balandligini kiriting : '); ReadLn(H);

Write(' Silindr asosi radiusini kiriting: '); ReadLn(R);

V := Pi \* R \* R \* H;

S := 2 \* Pi \* R \* H;

WriteLn('Silindr hajmi= ', V : 5 : 2); {Bunda 5 – V o'zgaruvchiinng qiymatini chop etishda umumiy xonalar soni, 2- esa kasr qismining xonalari soni}

WriteLn(' Silindr yon sirti yuzasi = ', S : 5 : 2);

ReadLn

END.

**1.2 - misol.** Tekislikda uchta nuqta berilgan.

*Ulardan qaysi biri koordinata boshiga yaqinligini aniqlang.*

### Test

Test tartibi	Berilganlar						Natija
	xA	yA	xB	yB	xC	yC	S
1	2	1	2	2	-1	3	"Bu A nuqta"
2	2	2	2	1	-1	3	"Bu B nuqta"
3	2	2	-1	3	2	1	"Bu C nuqta"

**Algoritmi:**

**Alg** Nuqta( **haq** xA,yA,xB,yB,xC,yC, **lit** S)

**arg** xA,yA,xB,yB,xC,yC

**natija** S

**boshl** haq DistA,DistB,DistC

```

DistA := sqrt(xA**2 + yA**2)
DistB := sqrt(xB**2 + yB**2)
DistC := sqrt(xC**2 + yC**2)
agar (DistA < DistB) va (DistA < DistC)
    u holda S := "Bu A nuqta"
    aks holda agar DistB < DistC
        u holda S := "Bu B nuqta"
        aks holda S := "Bu C nuqta"
hal bo' ldi
hal bo' ldi
tamom

```

Turbo Pascaldagi dasturi:

Program Points;

```

Var xA, yA, xB, yB, xC, yC, DistA, DistB, DistC : Real;
BEGIN
WriteLn('A nuqta koordinatasini kiriting:');
Write('x = '); ReadLn(xA); Write('y = '); ReadLn(yA);
WriteLn(' B nuqta koordinatasini kiriting:');
Write('x = '); ReadLn(xB); Write('y = '); ReadLn(yB);
WriteLn(' C nuqta koordinatasini kiriting:');
Write('x = '); ReadLn(xC); Write('y = '); ReadLn(yC);
DistA := sqrt(sqr(xA) + sqr(yA));
DistB := sqrt(sqr(xB) + sqr(yB));
DistC := sqrt(sqr(xC) + sqr(yC));
Write('S= ');
If (DistA < DistB) and (DistA < DistC)
then WriteLn(' Bu A nuqta.')
else If (DistB < DistC)
then WriteLn(' Bu B nuqta.')
else WriteLn(' Bu C nuqta.');
```

END.

**1.3 - misol.** Berilgan to'rt xonali butun sonning raqamlari ko'paytmasini toping.

Test

Test tartibi	Tekshirish	Son	Natija
1	Musbat son	2314	P = 24
2	Manfiy son	-1245	P = 40

**Algoritmi:**

**alg** Butun\_son (**but** Num, P)

**arg** Num

**natija** P

**boshl** butun i, j, k, l

```

Num := abs(Num)
i := Num div 1000
j := ((Num div 100) mod 10)
k := ((Num div 10) mod 10)
l := Num mod 10
P := i * j * k * l;

```

**tamom**

Turbo Pascaldagi dasturi:

```

Program Digit;
Var Number, i, j, k, l, P : Integer;
BEGIN
  ReadLn(Number);  Number:=Abs(Number);
  i := Number div 1000; Write(i:3);
  j := Number div 100 mod 10; Write(j:3);
  k := Number div 10 mod 10; Write(k:3);
  l := Number mod 10; WriteLn(l:3);
  P := i * j * k * l;  WriteLn( P);
  ReadLn
END.

```

**1.4 - misol.**  $ax^2 + bx + c = 0$  kvadrat tenglamani yeching.

Test

Test tartibi	Tekshirish	Koeffisientlar			Natija
		a	b	C	
1	$d > 0$	1	1	-2	$x_1 = 1, x_2 = -2$
2	$d = 0$	1	2	1	$x_1 = -1, x_2 = -1$
3	$d < 0$	2	1	2	Ildizlari mavjud emas
4	$A=0, b=0, c=0$	0	0	0	x ning ixtiyoriy qiymatida yechim nolga teng
5	$A=0, b=0, c \neq 0$	0	0	2	Noto'g'ri tenglama
6	$a=0, b \neq 0$	0	2	1	Chiziqli tenglama. $x = -0,5$ yagona yechim
7	$a \neq 0, b \neq 0, s = 0$	2	1	0	$x_1 = 0, x_2 = -0,5$

Algoritmi:

**alg** Kvad ( **haq** a, b, c, x1, x2, **lit** t)

**arg** a,b,c

**natija** x1,x2,t

**boshl** haq d

d :=  $b^2 - 4 * a * c$

**agar**  $d < 0$

**u holda**  $t := \text{"Haqiqiy ildizlari mavjud emas"}$   
**aks holda agar**  $d=0$   
     **u holda**  $t := \text{"Ildizlari"}; x1 := -b/(2*a); x2 := x1$   
     **aks holda**  $t := \text{"Ikkita ildizi"}$   
      $x1 := (-b + \text{sqrt}(d)) / (2*a)$   
      $x2 := (-b - \text{sqrt}(d)) / (2*a)$   
**hal bo' ldi**

**hal bo' ldi**

**tamom**

Turbo Pascaldagi dasturi:

Program Quadrat;

Var a, b, c : Real;

Discr : Real;

x1, x2 : Real;

BEGIN

  ReadLn(a, b, c);

  If (a=0) and (b=0) and (c=0)

  then begin Write('Hamma koefitsientlari nolga teng');

    WriteLn('x - ixtiyoriy son')

  end

  else

    If (a=0) and (b<>0)

    then WriteLn('Chiziqli tenglama. Yagona yechim: x =', -c/b:6:2)

    else

      If (a=0) and (b=0) and (c<>0)

      then WriteLn('Noto' ' g' ' ri tenglama.')

      else

      begin

        Discr := b\*b - 4\*a\*c;

        If Discr > 0

          then begin

$x1 := (-b + \text{Sqrt}(\text{Discr})) / (2*a);$

$x2 := (-b - \text{Sqrt}(\text{Discr})) / (2*a);$

            WriteLn('x1=', x1:6:2, ', x2=', x2:6:2)

          end

      else

        If Discr = 0

        then begin

$x1 := -b/(2*a);$

          WriteLn('Karrali ildizlar: x1=', x1:6:2, ' x2=', x1:6:2)

        end

```

else WriteLn('Haqiqiy ildizlari mavjud emas');
end;
ReadLn
END.

```

*Mustaqil ishlash uchun masalalar*

- 1.1. Radiuslari bir xil bo'lgan aylana uzunligi, doira yuzasi va sharning hajmini hisoblang.
- 1.2. Katetlari berilgan to'g'ri burchakli uchburchakning perimetri va yuzasini hisoblang.
- 1.3. Uchlarining koordinatalari berilgan uchburchakning perimetri va yuzasini hisoblang.
- 1.4. Uchta berilgan haqiqiy sonlar uchun o'rtacha geometrik qiymatini kasr qismini hisoblang.
- 1.5. Berilgan A butun sonni ikki xonali toq son ekanligini aniqlang.
- 1.6. Berilgan uchta sonning eng kattasini aniqlang.
- 1.7. Ikkita to'g'ri to'rtburchak o'zini tomonlari bilan berilgan. Birinchi to'g'ri to'rtburchak to'lig'icha ikkinchisining ichida joylashishini aniqlang.
- 1.8. Qiymatlari ma'lum bo'lgan a, b, c sonlarni shunday joylashtiringki a, b, c lar mos ravishda eng kichik, o'rtacha va eng katta qiymatli tartibda joylashsin.
- 1.9.  $ax^4 + bx^2 + c = 0$  bikvadrat tenglamani yeching.
- 1.10. Koordinatalari (x, y) berilgan nuqtaning qaysi chorakda yotganini aniqlang.

## 2. For (uchun) tipidagi sikl yordamida bajariladigan algoritm va dasturlar tuzish

For (uchun) tipidagi sikl oldindan takrorlashlar soni ma'lum bo'lganda ishlatiladi.

Til	Misol	Qadam qiymati
Algoritmik	<b>so</b> i uchun 1 dan N gacha sikl tanasi	Hamisha 1
Pascal	For i := 1 to N do sikl tanasi;	1
	For i := N downto 1 do sikl tanasi;	-1

**2.1 - Misol.** Sonli massiv  $A = (a_1, a_2, \dots, a_N)$  ning elementlarini yig'indisini hisoblang.

### Test

	Berilgan	Natija
N=5	A=(3, 5, -2, 6, 3)	S=15.0

Algoritmi:

```
alg Summa (but N,  
           haq jad A[1:N], haq S)  
arg N,A  
boshl but i  
  S:=0  
  sb i uchun 1 dan N gacha  
    S := S + A[i]  
  so  
tamom
```

Algoritmning bajarilishi

i	S
	0
1	$0 + a_1 = 0 + 3 = 5$
2	$a_1 + a_2 = 3 + 5 = 8$
3	$a_1 + a_2 + a_3 = 8 - 2 = 6$
4	$a_1 + a_2 + a_3 + a_4 = 6 + 6 = 12$
5	$a_1 + a_2 + a_3 + a_4 + a_5 = 12 + 3 = 15$

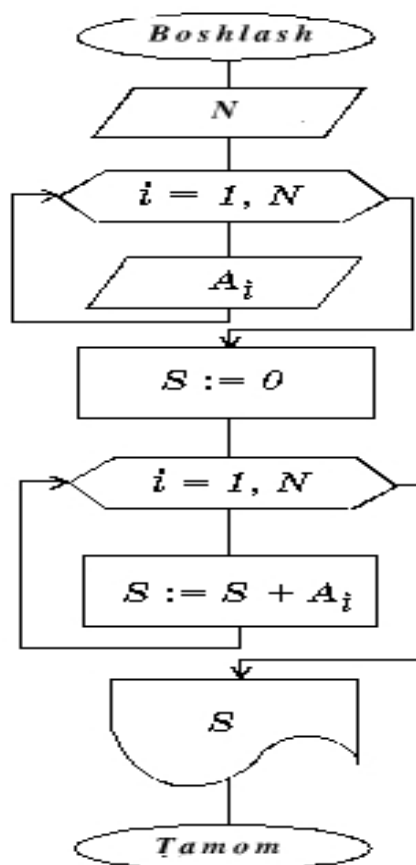
Turbo Pascaldagi dasturi:

```

Program Summa;
  Type Mas = Array [1..20] of Real;
  Var A    : Mas;
      i, N : Integer;
      S    : Real;
BEGIN
  ReadLn(N);
  For i := 1 to N do
  begin
    Write('A [', i, ' ] = ');
    ReadLn(A[i])
  end;
  S := 0;
  For i := 1 to N do S := S+A[i];
  WriteLn( S : 5 : 1);
  ReadLn
END.

```

Blok sxemasi:



2.2 - Misol. Berilgan  $A = (a_1, a_2, \dots, a_N)$  massivning eng katta qiymatini va uning nomerini aniqlang.

Test			
Berilganlar		Natija	
N=4	A=(3, -1, 10, 1)	Amax=10	K=3

Algoritmi:

**alg** Maks (**but** N, **haq** jad A[1:N], **haq** Amax, **but** k)

**arg** N, A

**natija** Amax, k

**boshl** but i

Amax := A[1]; k := 1

**sb** i **uchun** 2 **dan** N **gacha**

**agar** A[i] > Amax

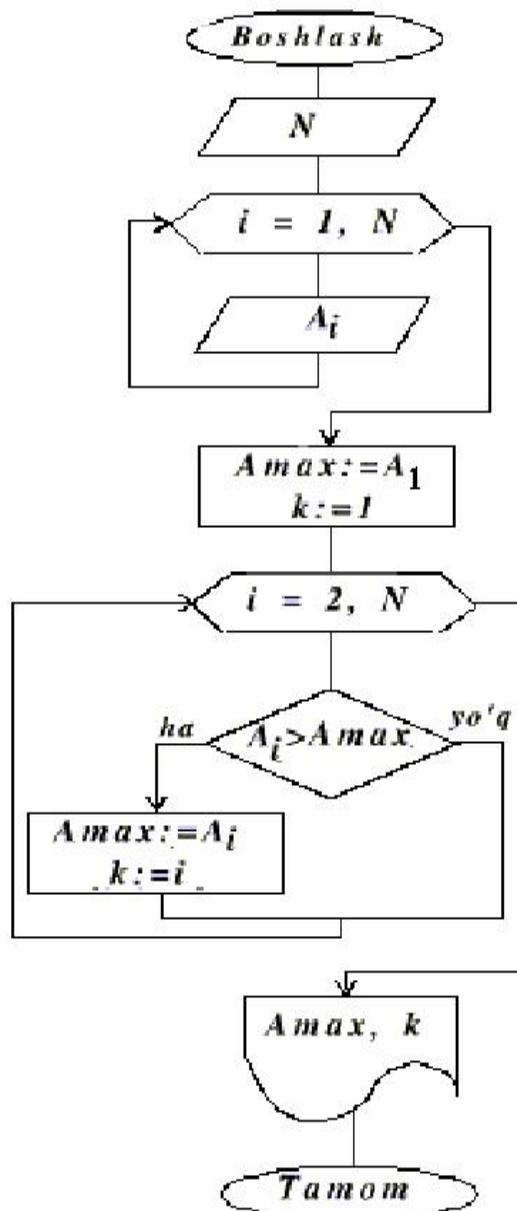
**u holda** Amax:=A[i]; k := i

**hal bo' ldi**

**so**

**tamom**

blok-sxemasi:



*Algoritmning bajarilishi*

I	A[I] > Amax	Amax	k
2	-	3	1
3	+	10	3
4	-		

*Turbo Pascaldagi dasturi:*

Program MaxElem;

Type Mas = Array [1..20] of Real;

Var A : Mas;

i, N : Integer;

k : Integer;

Amax : Real;

BEGIN

ReadLn(N);

For i := 1 to N do

begin



```

    Write('A [ ', i, ' ] = '); ReadLn(A[i])
end;
Amax := A[1]; k:=1;
For i := 2 to N do
    If A[i] > Amax then
        begin
            Amax := A[i]; k := i
        end;
WriteLn(k);
WriteLn(Amax : 5 : 1); ReadLn
END.

```

**2.3 - Misol.** *Sinfidagi o'quvchilar ichidan basketbol komandasi tuziladi. Komandaga bo'yi 170 dan yuqori bo'lgan o'quvchilar qabul qilinadi. Komandaga tushadigan davogarlarni ro'yxatini aniqlang.*

Test					
Test tartibi	Tekshirish	O'quvchilar soni	Familiyasi	Bo'yi	Natija
1	Nomzodlar bor	3	Kulov Chexin Uvarov	171 165 178	Kulov Uvarov
2	Nomzodlar yo'q	2	Ershov Ivanov	170 165	Nomzodlar yo'q

### Algoritmi:

```

alg Basketbol (but N, lit jad Fam[1:N], Nomz[1:N],
                haq jad Bo'yi[1:N] )
arg N, Fam, Bo'yi
boshl but i, k
    k:=0
sb i uchun 1 dan N gacha
    agar Bo'yi[i]>170
        u holda k:=k+1; Bo'yi [k] := Fam [i]
    hal bo'ldi
so
agar k=0
    u holda chiqarish "SINFDA NOMZODLAR YO' Q."
    aks holda sb i uchun 1 dan k gacha
        chiqarish Nomz[i]
    so
hal bo'ldi
tamom

```

*Algoritmnin bajarilishi*

Test tartibi	i	Bo'yi[i] > 170	K	Komandaga nomzodlar
1	1	+	0	Kulov Uvarov
	2	-	1	
	3	+	2	
2	1	-	0	-
	2	-		

TurboPascaldagi dasturi:

Program BascetBall;

Var

SurName : Array [1..30] of String;

Height : Array [1..30] of Real;

Nomz : Array [1..30] of String;

NPupil, i, K : Integer;

BEGIN ClrScr;

ReadLn(NPupil);

For i := 1 to NPupil do

begin Write(i, ' Familiyasi - '); ReadLn(SurName[i]);

Write(' Bo' yi - '); ReadLn(Height[i]);

end; WriteLn;

K:=0;

For i := 1 to NPupil do

If Height[i]>170 then

begin K:=K+1; Nomz[K] := SurName[i] end;

If K=0 then WriteLn('SINFDA NOMZODLAR YO' ' Q.')

else

begin WriteLn(' BASKETBOL KOMANDASIGA  
NOMZODLAR :');

For i := 1 to K do WriteLn( i, ' ', Nomz[i]);

end;

ReadLn

END.

**2.4 - Misol.**  $X(N)$  massiv berilgan. Yangi  $Y(N)$  massivi hosil qilingki unda  $X$  ning avval hamma musbat keyin nol undan keyin manfiy sonlari joylashgan bo'lsin.

Test	
Berilganlar	Natija
N=7 X=(-1, 2, 0, 4, -3,-2,0)	Y=(2, 4, 0, 0, -1, -3, -2)

Algoritmi

**alg** Yangi\_Tartib (but N, haq jad X[1:N], Y[1:N])

```

arg N, X
natija Y
boshl but i, k
    k := 0
sb i uchun 1 dan N gacha
    agar X[i] > 0
        u holda k := k+1; Y[k] := X[i]
    hal bo' ldi
so
sb i uchun 1 dan N gacha
    agar X[i] = 0
        u holda k := k+1; Y[k] := X[i]
    hal bo' ldi
so
sb i uchun 1 dan N gacha
    agar X[i] < 0
        u holda k := k+1; Y[k] := X[i]
    hal bo' ldi
so
tamom

```

Turbo Pascaldagi dasturi:

```

Program NewOrder;
Uses Crt;
Var N, i, k : Integer;    X, Y  : Array [1..20] of Real;
BEGIN
ClrScr; Write(' N = '); ReadLn(N);
For i := 1 to N do
    begin Write('X[ ', i, ' ] = '); ReadLn(X[i]) end;
k:=0;
For i := 1 to N do
    If X[i]>0 then    begin k:=k+1; Y[k]:=X[i] end;
For i := 1 to N do
    If X[i]=0 then
        begin k:=k+1; Y[k]:=X[i] end;

For i := 1 to N do
    If X[i]<0 then
        begin k:=k+1; Y[k]:=X[i] end;
For i := 1 to N do Write(Y[i] : 5 : 1);
WriteLn;
ReadLn
END.

```

*Mustaqil ishlash uchun masalalar*

2.1. Quyidagi ifodaning qiymatini hisoblang:

$$\begin{array}{l}
 \text{a)} \quad \sin x + \sin^2 x + \dots + \sin^n x ; \\
 \text{b)} \quad \sin x + \sin x^2 + \dots + \sin x^n ; \\
 \text{v)} \quad \sin x + \sin^2 x^2 + \dots + \sin^n x^n ;
 \end{array}$$

g)  $\sin x + \sin \sin x + \dots + \sin \sin \dots \sin x$  (n marta).

2.2. Berilgan  $A(N)$  massiv elementlarini  $B(N)$  massivga teskari tartib bilan yozing.

2.3.  $A(N)$  massiv berilgan.  $B(N)$  massivni quyidagi formula yordamida hosil qiling:  $b_i = (a_1 + a_2 + \dots + a_i) / i$ .

2.4. Quyidagini hisoblang:

$$P = 1 \cdot 2 + 2 \cdot 3 \cdot 4 + 3 \cdot 4 \cdot 5 \cdot 6 + \dots + N \cdot (N+1) \cdot \dots \cdot 2N.$$

2.5. Berilgan  $X(N)$  massivning maksimal komponentdan (agar ular bir nechta bo'lsa, maksimal komponentni tartibi bo'yicha birinchisini oling) oldingi barcha manfiy komponentlarni nol bilan almashtiring.

2.6. Berilgan  $X(N)$  massivning beshga karrali bo'lmagan barcha elementlari kvadratlari yig'indisini hisoblang.

2.9. Berilgan  $X(N)$  massivda eng kichik va eng katta elementlar o'rnini almashtiring.

2.10. Berilgan  $X(N)$  massivdagi berilgan qiymatdan kichiklarning sonini aniqlang.

2.11. Berilgan  $A(N)$  massivning komponentlarini chapga bir pozisiya siklik siljishni amalga oshiring, ya'ni  $A = (a_2, a_3, \dots, a_N, a_1)$  vektorni hosil qiling.

2.12. Berilgan  $A(N)$  massivning komponentlarini o'ngga ikki pozisiya siklik siljishni amalga oshiring, ya'ni  $A = (a_{N-1}, a_N, a_1, a_2, \dots, a_{N-2})$  vektorni hosil qiling.

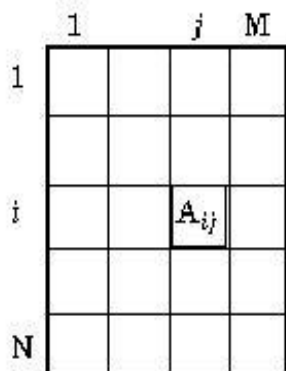
### 3. For (uchun) tipidagi ichma-ich joylashgan sikl yordamida bajariladigan algoritm va dasturlar

Til	For tipidagi ichma-ich joylashgan sikl sxemasi
-----	--

Algoritm tilda ifodalanishi	<b>sb i uchun A1 dan B1 gacha</b> tashqi sikl tanasi ..... <b>sb j uchun A2 dan B2 gacha</b> ichki sikl tanasi ..... <b>so</b> ..... <b>so</b>
Pascalda ifodalanishi	For i := A1 to B1 do begin ..... For j := A2 to B2 do begin ..... end; ..... end;

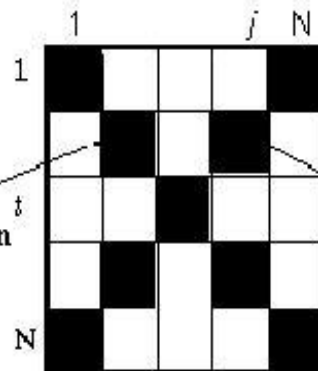
Ichma-ich sikllar ko'pincha matritsalar (ikki o'lchovli massiv, to'g'ri to'rtburchakli jadval) va vektorlarni (bir o'lchovli massiv, chiziqli jadval) qayta ishlashda ishlatiladi:

**A(N,M) matritsa**  
(to'g'ri to'rtburchakli)



$A_{ij}$  elementlar bilan matritsa ning bosh diagonali

**A(N,N) matritsa**  
(kvadrat)



$A_{i, N+i-i}$  elementlar bilan matritsa ning yon diagonali

3.1 - misol. Berilgan  $A(N, M)$  matritsaning ustunlari elementlarning yig'indisini hisoblang.

Test

Berilganlar		Natija
N=2 M=2	$A = \begin{pmatrix} 2 & 1 \\ 4 & 3 \end{pmatrix}$	S=(6,4)

Algoritmi:

Blok sxemasi:

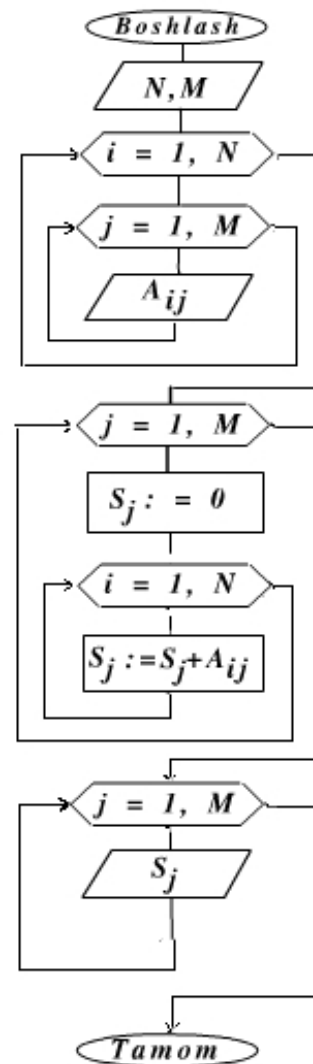
```

alg Ustun_Yig (but N, M,
    haq jad A[1:N, 1:M], haq jad S[1:M])
arg N,M,A
boshl but i, j
    sb j uchun 1 dan M gacha
        S[j]:=0
    sb i uchun 1 dan N gacha
        S[j]:=S[j] + A[i, j]
    so
so
tamom

```

*Algoritmning bajarilishi*

j	i	S[i]
1	1	S1=0
	2	S1=0+2=2 S1=2+4=6
2	1	S2=0
	2	S2=0+1=1 S2=1+3=4



*Turbo Pascaldagi dasturi:*

```

Program SumColumn;
Var      A: Array [1..10, 1..10] of Real;
        N, M, i, j : Integer;
        S: Array [1..10] of Real;
Begin
Write('Satriklar soni - '); ReadLn(N);
Write('Ustunlar soni - '); ReadLn(M);
For i := 1 to N do
For j := 1 to M do
begin Write('A[', i, ', ', j, ']= ? ');
      ReadLn(A[i, j])
WriteLn(' A matritsa');
For i := 1 to N do
begin
For j := 1 to M do Write(A[i, j] : 5 : 1);
WriteLn
end;
end;

```

```

For j := 1 to M do
  begin  S[j] := 0;
    For i := 1 to n do S[j] := S[j] + A[i, j]
    end;
  Write( 'J A V O B: Ustun elementlari yig'indisi teng');
  For j := 1 to M do Write(S[j] : 5 : 1); ReadLn
END.

```

**3.2 - misol.** Berilgan butun sonli  $A(N, M)$  matritsada maksimal qiymati necha marta uchrashishini aniqlang.

Test		
Berilgan		Natija
N=2 M=3	$A = \begin{pmatrix} 1 & 2 & 5 \\ 5 & 1 & 5 \end{pmatrix}$	K=3

Algoritmi:

**alg** Maks\_soni(**but** N,M,K **but jad** A[1:N,1:M])

**arg** N,M,A

**nat** K

**boshl but** i, j, Amax

Amax := A[1, 1]

**sb** i uchun 1 dan N gacha

**sb** j uchun 1 dan M gacha

**agar** A[i, j] > Amax

**u holda** Amax := A[i, j]

**hal bo'ldi**

**so**

**so**

K := 0

**sb** i uchun 1 dan N gacha

**sb** j uchun 1 dan M gacha

**agar** A[i, j] = Amax

**u holda** K := K+1

**hal bo'ldi**

**so**

**so**

**tamom**

<u>Algoritmning bajarilishi (Amax ni topish)</u>				(Amax necha marta uchrashini aniqlash)			
i	j	A[i,j]>Amax	Amax	i	j	A[i,j]=Amax	K
			1	1	1	-	0
1	1	-			2	-	

	2	+	2					
	3	+	5		2	3	+	1
2	1	-			1		+	2
	2	-			2		-	
	3	-			3		+	3

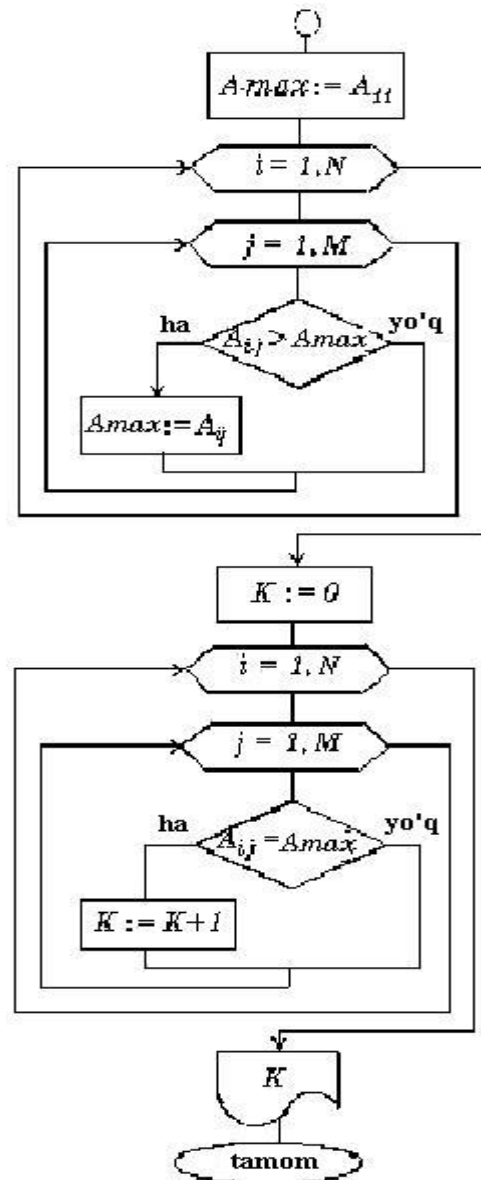
Blok-sxemasi fragmenti:

### Turbo Pascaldagi dasturi:

```

Program NumberOfMaximums;
Type Mas =Array[1..10,1..10] of Integer;
Var A: Mas;
    N, M, K, Amax, i, j : Integer;
Begin
    Write('Satrlar soni - '); ReadLn(N);
    Write('Ustunlar soni - '); ReadLn(M);
    For i := 1 to N do
        For j := 1 to M do
            begin Write('A[', i, ', ', j, ']=? ');
                ReadLn(A[i, j])
            end; WriteLn;
        WriteLn(' A Matritsa');
        For i := 1 to N do
            begin For j := 1 to M do Write(A[i, j] : 5);
                WriteLn
            end;
            Amax := A[1, 1];
            For i := 1 to N do
                For j := 1 to M do
                    if A[i, j] > Amax then Amax := A[i, j];
                    K:=0;
                    For i := 1 to N do
                        For j := 1 to M do
                            if A[i, j] = Amax then K := K+1;
                    WriteLn(Amax:3, ' maksimal soni ', K,
                        ' marta uchraydi');
                ReadLn;
            END.

```



**3.3 - misol.** Berilgan  $A(N, M)$  matritsaning  $P$  va  $Q$  satrlari o'rnini almashtiring ( $1 \leq P \leq N, 1 \leq Q \leq M$ ).

### Test

Berilgan	Natija
----------	--------



$N=3 \ M=3 \ P=1 \ Q=3$ $A = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 2 & 2 \\ 3 & 1 & 3 \end{pmatrix}$	$A = \begin{pmatrix} 3 & 1 & 3 \\ 2 & 2 & 2 \\ 1 & 2 & 1 \end{pmatrix}$
--	---

*Algoritmi:*

**alg** Almashtirish(**but** N, M, P, Q, **haq jad** A[1:N, 1:M] )

**nat** N,M,A,P,Q

**natija** A

**boshl but** j, **haq** Tmp

**sb j uchun 1 dan M gacha**

Tmp:=A[P, j]; A[P, j]:=A[Q, j]; A[Q, j]:=Tmp

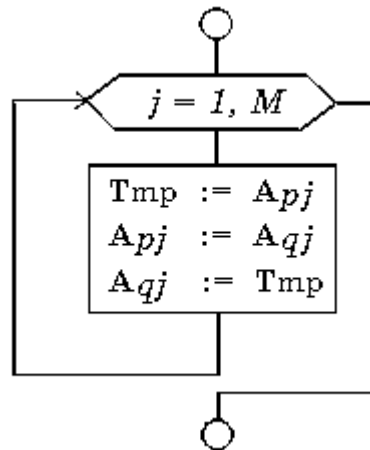
**so**

**tamom**

*Algoritmnin bajarilishi*

j	Tmp	A[1,j]	A[3,j]
1	1	3	1
2	2	1	2
3	1	3	1

*Blok-sxemasi fragmenti:*



*Turbo Pascaldagi dasturi:*

Program Exchange;

Type Mas = Array [1..10, 1..10] of Real;

Var A : Mas; N, M, P, Q, i, j : Integer; Tmp : Real;

Begin

  ReadLn(N,M);

  For i := 1 to N do

    For j := 1 to M do

      begin Write('A[', i, ', ', j, '] = ? ');

        Read(A[i, j])

      end; WriteLn;

  Write('P = '); ReadLn(P); Write('Q = '); ReadLn(Q);

  For i := 1 to N do

    begin

      For j := 1 to M do Write(A[i, j] : 5 : 1);

      WriteLn

    end;

  For j := 1 to M do

    begin Tmp:=A[P, j]; A[P, j]:=A[Q, j]; A[Q, j]:=Tmp end;

```

WriteLn('Natija matritsa:');
For i := 1 to N do
begin
  For j := 1 to M do Write(A[i, j] : 5 : 1) ;
  WriteLn
end; ReadLn
END.

```

**3.4 - misol.**  $a_1, a_2, \dots, a_N$  ketma-ketlikni o' sish tartibida joylashtiring.

Berilgan		Natija
N=4	A=(5, 2, 7, 1)	A=(1, 2, 5, 7)

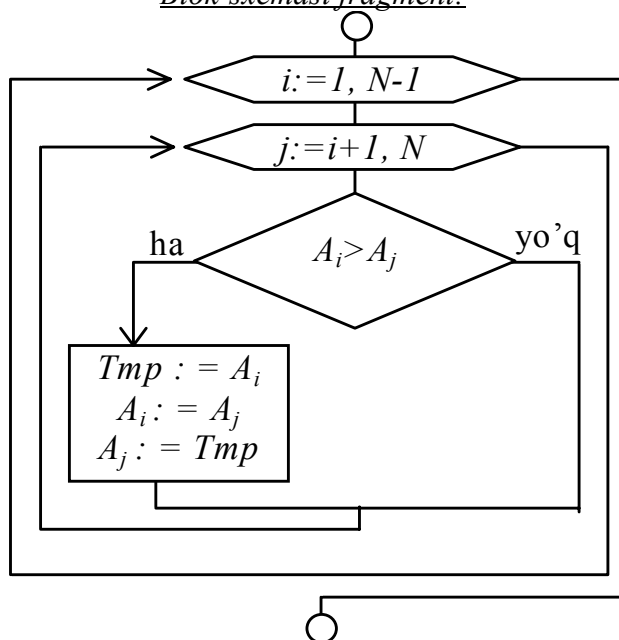
Algoritmi:

```

alg O' sish (but N, haq jad A[1:N])
arg N,A
natija A
boshl but i, j, xak Tmp
  sb i uchun 1 dan N-1 gacha
    sb j uchun i+1 dan N gacha
      agar A[i] > A[j] u xolda
        Tmp:=A[i]; A[i]:=A[j]; A[j]:=Tmp
      hal bo' ldi
    so
  so
tamom

```

Blok-sxemasi fragment:



Algoritmning bajarilishi

I	J	A[i]>A[j]	A massiv
1	2	+	2, 5, 7, 1
	3	-	
	4	+	1, 5, 7, 2
2	3	-	
	4	+	1, 2, 7, 5
3	4	+	1, 2, 5, 7

Turbo Pascaldagi dasturi:

```

Program Regulation;
Type Mas = Array [1..10] of Real;
Var A : Mas; i, j, N : Integer; Tmp : Real;
Begin
  ReadLn(N);
  For i := 1 to N do

```

```

begin Write('A [ ' , i , ' ] = '); ReadLn(A[i]) end;
For i := 1 to N-1 do
  For j := i+1 to N do
    If A[i] > A[j] then
      begin Tmp:=A[i]; A[i]:=A[j]; A[j]:=Tmp end;
  For i:=1 to N do Write( A[i] : 6 : 1);
WriteLn; ReadLn

```

END.

**3.5-misol**  $A(N, N)$  massivdagi bosh digonaldan yuqorida joylashgan elementlar va pastda joylashgan elementlar yig'indilarini hisoblang.

Test	
Berilgan	Natija
$N=3$ $\begin{pmatrix} 1 & 2 & 4 \\ 3 & 1 & 3 \\ 2 & 1 & 2 \end{pmatrix}$	$S_1=6$ $S_2=9$

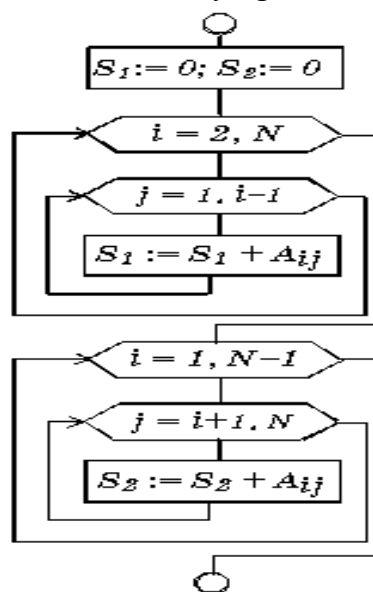
Algoritmi:

```

alg Yig' (but N,
haq jad A[1:N, 1:N], haq S1, S2)
arg N,A natija S1,S2
boshl but i, j
  S1:=0; S2:=0
  sb i uchun 2 dan N gacha
    sb j uchun 1 dan i-1 gacha
      S1:=S1 + A[i, j]
    so
  so
  sb i uchun 1 dan N-1 gacha
    sb j uchun i+1 dan N gacha
      S2:=S2 + A[i, j]
  so
so
tamom

```

Blok-sxemasi fragmenti:



Algoritmnining bajarilishi

i	j	S1	S2
		0	0
2	1	0+3=3	
3	1	3+2=5	
	2	5+1=6	

1	2		0+2=2
	3		2+4=6
2	3		6+3=9

Turbo Pascaldagi dasturi:

Program TwoSums;

Var A : Array [1..10, 1..10] of Real;

S1, S2 : Real;

N, i, j : Integer;

{-----}

Begin

ReadLn(N);

For i := 1 to N do

For j := 1 to N do

begin Write('A[', i, ', ', j, '] = ? ');

ReadLn(A[i, j])

end;

WriteLn(' A Matritsa ');

For i := 1 to N do

begin

For j := 1 to N do Write(A[i, j] : 5 : 1); WriteLn

end;

S1 := 0;

For i := 2 to N do

For j := 1 to i-1 do

S1 := S1 + A[i, j];

WriteLn('Javob :');

WriteLn(S1:5:1);

End;

S2 := 0;

For i := 1 to N-1 do

For j := i+1 to N do

S2 := S2 + A[i, j];

WriteLn(S2:5:1);

ReadLn

END.

### *Mustaqil ishlash uchun masalalar*

**3.1.** Berilgan  $A(N, M)$  matritsada eng katta elementni va u joylashgan satr hamda ustun nomerini toping.

**3.2.** Berilgan  $A(N, M)$  matritsada har bir satr elementlarining yig'indisini, musbatlari o'rta arifmetigi va sonini hisoblang

**3.3.** Berilgan  $A(N, M)$  butun sonli matritsaning elementlari yig'indisi juft son bo'ladimi, yo'kmi aniqlang.

**3.4.** Berilgan  $A(N, M)$  matritsada barcha elementlarining o'rta arifmetigidan katta bo'lgan elementlar sonini aniqlang.

3.5. Berilgan  $A(N, M)$  butun sonli matritsaning toq qiymatli elementlarining yig'indisi va ko'paytmasini hisoblang.

3.6.  $A(N, M)$  matritsa berilgan.  $X(M)$  vektorni hisoblang, bu erda  $X_j$  qiymati  $A$  matritsaning  $j$ -chi usundagi musbat elementlar yig'indisi.

3.8.  $A(N, M)$  matritsa berilgan.  $X(M)$  vektorni hosil qiling, u matritsaning P-satriga teng va  $Y(N)$  vektorni hosil qiling, u matritsaning  $Q$ -ustiniga teng.

3.9. Berilgan  $A(N, M)$  matritsadagi eng katta va eng kichik elementlari o'rnini almashtiring.

3.10. Berilgan  $n$  natural songa ko'ra  $(n, n)$  tartibli ushbu shakl-dagi matritsalarini hosil qiling.

A) $\begin{matrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{matrix}$	b) $\begin{matrix} n & n-1 & n-2 & \dots & 1 \\ 0 & n & n-1 & \dots & 2 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & n \end{matrix}$	v) $\begin{matrix} 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & \dots & 1 & 2 \\ \dots & \dots & \dots & \dots & \dots \\ 1 & 2 & \dots & n-1 & n \end{matrix}$
---	---	---

#### 4. While (toki) tipli sikl yordamida bajariladigan algoritm va dasturlar tuzish

**Toki** tipidagi sikllar yordamida algoritmning istalgan takrorlanuvchi qismini dasturlash mumkin. Ammo, amaliyotda **toki** tipidagi sikl quyidagi ikki holatda ko'prok ishlatiladi:

- **Takrorlanish soni oldindan noma'lum** (masalan, natijani talab qilingan aniqlikka erishishigacha bo'lgan, massivning birinchi manfiy elementigacha bo'lgan sikl va sh.k.). Bunday sikl **uzlikli toki** tipidagi sikl deb ataladi.
- **Takrorlanish soni olinlan ma'lum, ammo sikl parametr kadami 1 ga** (maktab algoritmik tilida) **yoki 1 ga, -1 ga** (Pascal da) teng emas. Bunday sikl **uzliksiz toki** tipidagi sikl deb ataladi.

Uzlikli toki tipidagi sikl

Til	Misol	Izoh
Algoritmik tilda ifodalanishi	$i:=1$ ; Flag:="Yo'q" <b>sb toki</b> ( $i \leq N$ ) <b>va</b> (Flag="Yo'q")	Masala: $A(N)$ massivning birinchi manfiy element nomerini aniqlash. Bu erda Flag

	<b>agar</b> $A[i] < 0$ <b>u holda</b> $Flag := "Xa"; k := i$ <b>aks holda</b> $i := i + 1$ <b>hal bo' ldi</b> <b>so</b>	— "boshqaruvchi" liter tipli o'zgaruvchi ( mantiqiy yoki butun tiplardan ham foydalanish mumkin).
Pascal tilda ifodalanishi	$i := 1; Flag := FALSE;$ <b>While</b> $(i \leq N)$ <b>and not</b> $Flag$ <b>do</b> <b>If</b> $A[i] < 0$ <b>then begin</b> $Flag := TRUE; k := i$ <b>end</b> <b>else</b> $i := i + 1;$	Bu erda $Flag$ — $TRUE$ (chin) yoki $FALSE$ (yolg'on) qiymatlarni qabul qiluvchi mantiqiy o'zgaruvchi, <b>and</b> - 'va' operatsiyasi, <b>not</b> - 'emas' operatsiyasi.

uzliksiz toki tipidagi sikl

Til	Misol	Izoh
Algoritmik tilida ifodalanishi	$i := 1; S := 0$ <b>sb toki</b> $i \leq N$ $S := S + A[i]$ $i := i + 2$ <b>so</b>	$A(N)$ massivning toq indeksli elementlari yig'indisini hisoblash. Bunday elementlar soni oldindan ma'lum. Sikl parametr qadami ikkiga teng.
Pascal tilida ifodalanishi	$i := 1; S := 0;$ <b>While</b> $i \leq N$ <b>do</b> <b>begin</b> $S := S + A[i];$ $i := i + 2$ <b>end;</b>	

Toki tipidagi sikllarni tashkillashtirishda quyidagidan ham foydalanish mumkin:

- **Pascal** tilidagi **Repeat...until** so'ng shartli sikl operator:

Repeat <b>sikl tanasi</b> until <shart>	Siklni yakunlash sharti bajarilguncha <b>sikl tanasi</b> bajarilishi takrorlanadi.
---	--

**4.1 - misol.**  $a_1, a_2, \dots, a_N$  sonlar ketma-ketligi monoton kamayuvchi bo'lishligini aniqlash.

## Test

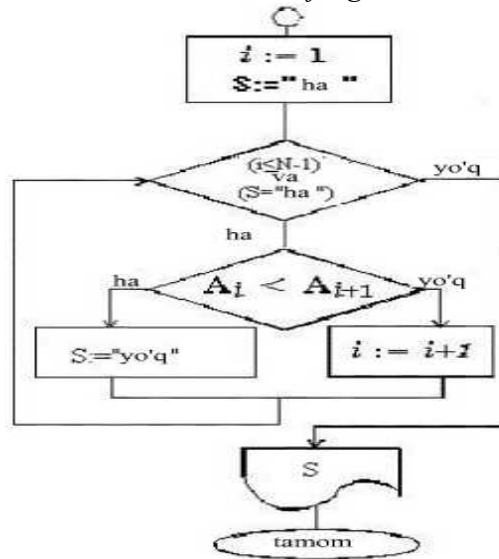
Nomer	Tekshirish	Berilgan		Natija
		N	A- vektor	
1	Bo'ladi	3	(3, 2, 1)	'Ha'

2	Bo'lmaydi	3	(2, 3, 1)	'Yo'q'
---	-----------	---	-----------	--------

Algoritmi:

alg kamayish ( but N,  
 haq jad A[1:N], lit S)  
 arg N, A  
 natija S  
 boshl but i  
 i:=1; S:="Ha"  
 sb toki (i<=N-1) va (S="Ha")  
 agar A[i] < A[i+1]  
 u holda S := "Yo'q"  
 aks holda i:=i+1  
 hal bo'ldi  
 so  
 tamom

Blok-sxemasi fragmenti:



Algoritmning bajarilishi

Tekshirilayotgan shartning belgilanishi:

$$(i \leq N-1) \text{ va } (S = \text{"Ha"}) \Rightarrow (1)$$

$$A[i] < A[i+1] \Rightarrow (2)$$

test	i	S	(1)	(2)
1	1	"Ha"	+	-
	2		+	-
	3		-(so)	
2	1	"Ha" "Yo'q"	+	+

Turbo Pascaldagi dasturi

```

Program Decrease;
  Var A      : Array [1..10] of Real;
      N, i : Integer;
      S: Boolean;
Begin
  ReadLn(N);
  For i := 1 to N do
    begin Write(' A[', i, '] = ');
          ReadLn(A[i])
    end
  
```

```

end;
For i := 1 to N do Write(A[i] : 5 : 1);
WriteLn
S := TRUE; i:=1;
While (i<=N-1) and S do
  If (A[i]<A[i+1]) then S := FALSE
                    else i := i+1;
WriteLn(' Berilgan ketma-ketlik ');
If S then Write(' manoton kamayuvchi ' )
        else Write(' manoton kamayuvchi emas ' );
ReadLn
END.

```

**4.2-misol.** *Tekislikda nuqtalar to'plami berilgan. Markazi (a, b) nuqtada va radiusi R aylana bilan chegaralangan sohada hech bo'lmaganda bitta nuqta yotishligini aniqlang.*

### Test

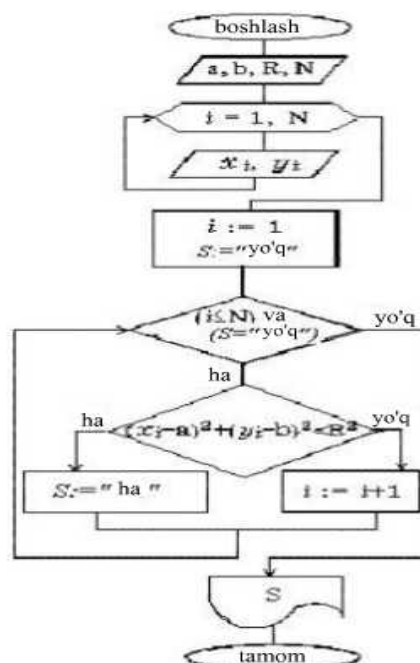
Nomer	Tekshirish	Berilgan					Natija
		a	b	R	Nuqtalar soni	Nuqtalar koordinatasi	S
1	Tegishli	1	0	2	3	X=(-1, 2, 3) Y=(2, 1, 2)	"Ha"
2	Tegishli emas	1	0	2	2	X=(-1, 3) Y=(2, 2)	"Yo'q"

Algoritmi:

Blok-sxemasi:



**alg** Nukta (**but** N,  
**haq jad** X [1 : N] , Y [1 : N] ,  
**haq** a, b, R, **lit** S)  
**arg** N,X,Y  
**natija** S  
**boshl but** i  
    i:=1; S:="Yo'q"  
    **sb toki** (i<=N) **va** (S="Yo'q")  
        **agar** (X[i]-a)\*\*2+(Y[i]-b)\*\*2  
        <R\*R  
            **u holda** S := "Xa"  
            **aks holda** i:=i+1  
    **hal bo' ldi**  
**so**  
**tamom**



Algoritmning bajarilishi

Tekshirilayotgan shartning belgilanishi:

$$(i \leq N) \text{ va } (S = \text{"Yo'q"}) \Rightarrow (1)$$

$$(X[i]-a)^2 + (Y[i]-b)^2 < R^2 \Rightarrow (2)$$

Test №	i	S	(1)	(2)
1	1	"Yo'q"	+	-
	2	"Ha"	+	+
	3		-(so)	
2	1	"Yo'q"	+	-
	2		+	-
	3		-(so)	

Turbo Pascaldagi dasturi:

```

Program SetOfPoints;
Type Mas = Array [1..20] of Real;
Var X, Y          : Mas;
    i, NPoints    : Integer;
    a, b, Radius  : Real;
    Flag          : Boolean;

Begin
ReadLn(a, b);          ReadLn(Radius);  ReadLn(NPoints);
For i := 1 to NPoints do
begin
WriteLn(i : 4, ' -chi nuqta' );
Write(' X = '); ReadLn(X[i]);

```

```

Write(' Y = '); ReadLn(Y[i]);
end;
WriteLn; Flag := FALSE ; i := 1;
While (i<=NPoints) and not Flag do
  If Sqr(X[i]-a)+Sqr(Y[i]-b) < Sqr(Radius)
    then Flag := TRUE
    else i:=i+1;
Write(' Javob: Nuqtalarning sohaga tegishli bo' ' lganlari' );
If Flag then
  WriteLn(' uchraydi' )
  else
  WriteLn(' uchramaydi' );
ReadLn
END.

```

**4.3 - misol.** Berilgan  $A(N, N)$  butun sonli matritsaning bosh diagonali elementlari ichida hech bo'lmaganda bitta toq musbat element mavjudmi aniqlang.

**Test**

test	Tekshirish	Berilgan		Natija
		N	A matritsa	Tekst
1	mavjud	3	$\begin{pmatrix} 2 & 2 & 2 \\ 2 & 3 & 2 \\ 2 & 2 & 5 \end{pmatrix}$	"Bundaylar bor "
2	mavjud emas	2	$\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$	" Bundaylar yo'q "

**Algoritmi**

**alg** Diagonal (**but** N, **but jad** A[1:N, 1:N], **lit** Tekst)

**arg** N,A

**natija** Tekst

**boshl** but i, **lit** Flag

i:=1; Flag:="Yo'q"

**sb toki** (i<=N) va (Flag="Yo'q")

**agar** (A[i, i]>0) va (A[i, i] mod 2=1)

**u holda** Flag := "Ha"

**aks holda** i:=i+1

**hal bo'ldi**

**so**

**agar** Flag = "Ha"

**u holda** Tekst := "Bundaylar bor"

**aks holda** Tekst := "Bundaylar yo'q"  
**hal bo'ldi**

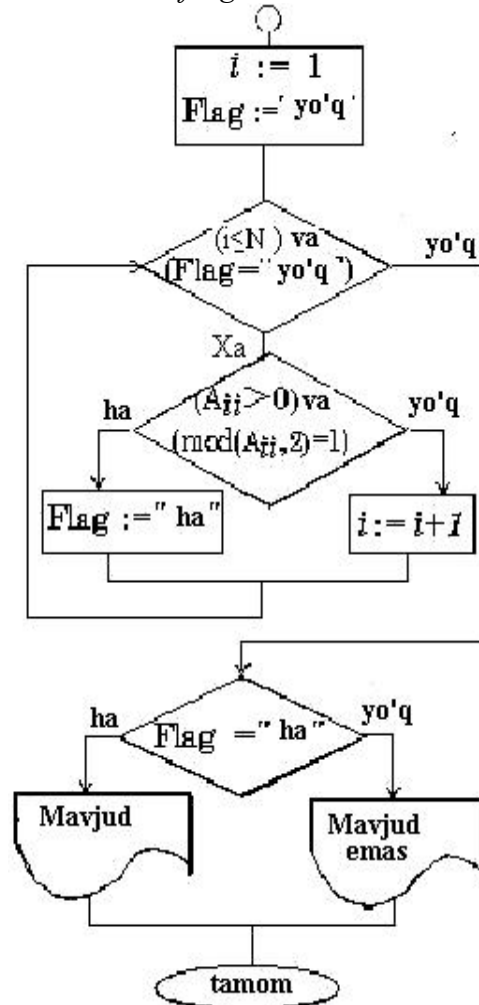
**tamom**

*Algoritmning bajarilishi*

Tekshirilayotgan shartning belgilanishi:  
 $(i \leq N) \text{ va } (\text{Flag} = \text{"Yo'q"}) \Rightarrow (1)$   
 $(A[i, i] > 0) \text{ va } (A[i, i] \bmod 2 = 1) \Rightarrow (2)$

test	I	Flag	(1)	(2)	Tekst
1	1	"Yo'q"	+	-	"Bundaylar bor "
	2	"Ha"	+	+	
2	1	"Yo'q"	+	-	" Bundaylar yo'q "
	2		+	-	
	3		-(so)		

*blok-sxemasi fragmenti:*



### Turbo Pascaldagi dasturi

Program Diagonal;

Type Mas = Array [1..10, 1..10] of Integer;

Var A : Mas;

N, i, j : Integer;

Flag : Boolean;

{-----}

Begin

For i := 1 to N do

For j := 1 to N do

begin

Write(' A[', i, ', ', j, ' ] = ? ');

ReadLn(A[i, j])

end;

For i := 1 to N do

```

begin
  For j := 1 to N do Write(A[i, j] : 5);      WriteLn
end;
{-----}
  Flag:=FALSE;
  i:=1;
  While (i<=N) and not Flag do
    If (A[i, i]>0) and (A[i, i] mod 2 = 1)
      then Flag:=TRUE
      else i:=i+1;
  WriteLn(' Javob : ' );
  Write(' Bosh diagonal elementlari ichida ');
  If Flag then WriteLn(' toq manfiylar bor. ')
  else WriteLn(' toq manfiylar yo' ' q. ');
  ReadLn;
END.

```

4.4 - Misol. *Fibonachchi* ( $F_i$ ) sonlari  $i = 2, 3, ..$  uchun  $F_0 = F_1 = 1$ ;  $F_i = F_{i-1} + F_{i-2}$  formula bo'yicha aniqlanadi (har navbatdagi son oldingi ikkitasining yig'indsiga teng). Berilgan sondan oshmaydigan *Fibonachchi* sonlarining yig'indisini hisoblang.

**Test**

Test nomeri	Berilganlar	Natija
1	M=10	S=1+1+2+3+5+8=20
2	M=1	S=1+1=2

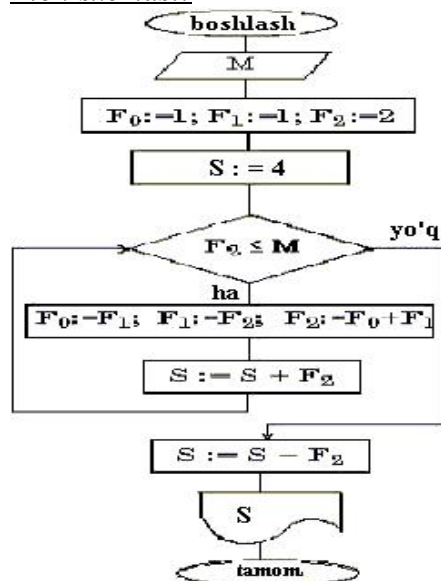
Algoritmi:

```

alg Fibonachchi (but M, S)
arg M
natija S
boshl but F0, F1, F2
  F0:=1; F1:=1; F2:=2
  S:=4
sb toki F2<=M
  F0:=F1; F1:=F2; F2:=F0+F1
  S:=S+F2;
so
  S:=S-F2
tamom

```

Blok-sxemasi:



Algoritmning bajarilishi

F0	F1	F2	S	F2<M
1	1	2	4	ha
1	2	3	7	ha
2	3	5	12	ha
3	5	8	20	yo'q

1	1	2	4	+
1	2	3	4+3=7	+
2	3	5	7+5=12	+
3	5	8	12+8=20	+
5	8	13	20+13=33	-(so)
			33-13=20	

Turbo Pascaldagi dasturi

Program SummaFib;

Var M, F0, F1, F2, S : Integer;

BEGIN

  ReadLn(M);

  F0:=1; F1:=1; F2:=2;

  S:=4;

  Write( M, ' :', F0:4, F1:4);

  While F2<=M do

    begin

      F0:=F1; F1:=F2; Write(F1 : 4);

      F2:=F0+F1; S:=S+F2;

    end;

  S:=S-F2;

  WriteLn(S); ReadLn

END.

4.5- Misol.  $A(N)$  massiv elementlari o' sish tartibida, uning tarkibiga tartibini buzmaganda holda berilgan  $D$  sonini kiriting.

Test

Test	Tekshirish	Berilganlar		Natija
		D	A massiv	
1	$D \leq a_1$	0	$A=(1, 3, 5)$	$A=(0, 1, 3, 5)$
2	$a_1 < D \leq a_N$	4	$A=(1, 3, 5)$	$A=(1, 3, 4, 5)$
3	$a_N < D$	6	$A=(1, 3, 5)$	$A=(1, 3, 5, 6)$

Algoritmi:

**alg** qo' shish (but N, haq D, haq jad  $A[1:N+1]$ )

**arg** N,D,A

**natija** A

**boshl** but i

$i := N$

**sb toki ( $i \geq 1$ ) va ( $A[i] > D$ )**

$A[i+1] := A[i]$

$i := i - 1$

**so**

$A[i+1] := D$

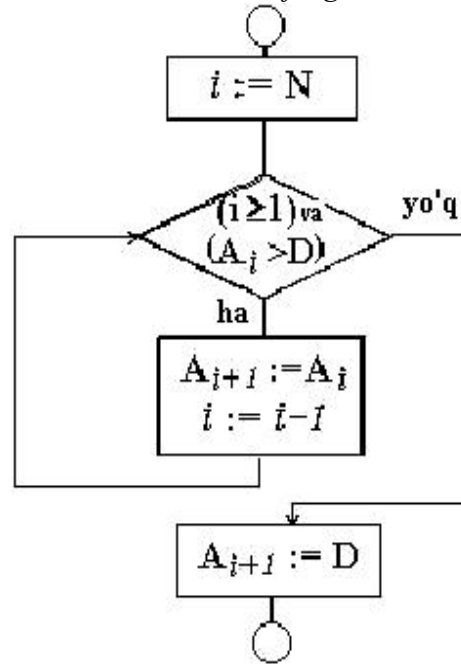
**tamom**

Algoritmning bajarilishi

Tekshirilayotgan shartning belgilanishi: ( $i \geq 1$ ) va ( $A[i] > D$ )  $\Rightarrow$  (1)

test	I	(1)	A massiv
1	3	+	(1, 3, 5)
	2	+	(1, 3, 5, 5)
	1	+	(1, 3, 3, 5)
		-(so)	(1, 1, 3, 5)
			(0, 1, 3, 5)
2	3	+	(1, 3, 5)
	2	-(so)	(1, 3, 5, 5)
			(1, 3, 4, 5)
3	3	-(so)	(1, 3, 5)
			(1, 3, 5, 6)

Blok-sxemasi fragmenti:



Turbo Pascaldagi dasturi:

Program Insertion;

Var A : Array [1..20] of Real;

D : Real;

N, i : Integer;

Begin

ReadLn(N);

For i := 1 to N do

begin Write(' A[', i, '] = '); ReadLn(A[i]) end;

ReadLn(D);

For i := 1 to N do Write(A[i] : 5 : 1);

WriteLn(D : 5 : 1);

i:=N;

While (i >= 1) and (A[i] > D) do

begin A[i+1] := A[i]; i:=i-1 end;

A[i+1] := D

For i := 1 to N+1 do Write( A[i] : 5 : 1); WriteLn;

ReadLn

END.

### **Mustaqil ishlash uchun masalalar**

4.1.  $Z = 1 + 2 + 3 + \dots$  yig'indini hisoblang. Hisoblashni  $Z$  qiymati berilgan  $A$  qiymatdan oshganda to'xtating.

4.2. Berilgan  $a_1, a_2, \dots, a_N$  butun sonlar ketma-ketlikining nolga teng elementlari borligini tekshiring. Agar bor bo'lsa, ulardan birichisini nomerini toping, yo'q bo'lsa, mos tekstni chop eting.

4.3. Berilgan  $A(N)$  vektorda ikkita ketma-ket keluvchi nol element mavjudmi tekshiring.

4.4. Berilgan  $A(N)$  vektorda uchta ketma-ket keluvchi bir xil ishorali element mavjudmi tekshiring.

4.5. Fazoda nuqtalar to'plami o'zining butun qiymatli koordinatalari bilan berilgan. nuqtalar hech bo'lmaganda bittasi koordinata boshi bilan mos tushadimi tekshiring.

4.6.  $A > 1$  butun son berilgan.  $5^k > A$  shartni qanoatlantiruvchi eng kichik  $k$  nomanfiy butun sonni toping.

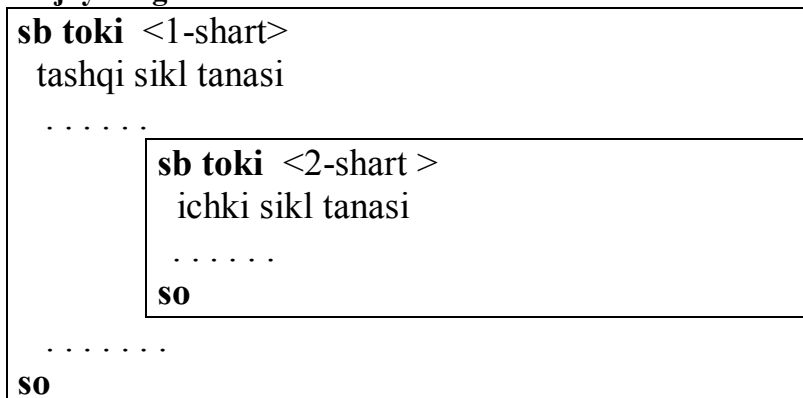
4.7.  $N$  – natural son berilgan.  $U$  nechta raqamdan tuzilgan tekshiring

4.8. Berilgan natural sonning raqamlar yig'indisini toping.

4.9. Berilgan natural sonning raqamlarini teskari tartibda yozing.

### **5. TOKI tipidagi ichma-ich joylashgan sikl yordamida bajariladigan algoritm va dasturlar**

Toki tipidagi ichma-ich joylashgan sikl sxemasi:



5.1 - misol. Berilgan butun qiymatli  $A(N)$  massivda hech bo'lmaganda bitta son qiymati bo'yicha mos tushuvchi juftlik bormi aniqlang.

#### **Test**

Test	Tekshirish	Berilgan		Natija
		N	A massiv	S
1	Mavjud	4	(1,3,2,3)	"Mos tushuvchi son bor"
2	Mavjud emas	3	(1,2,3)	" Mos tushuvchi son yo'q "

Algoritmi:

alg moslik(but N, but jad A[1:N], lit S)

arg N, A

nat S

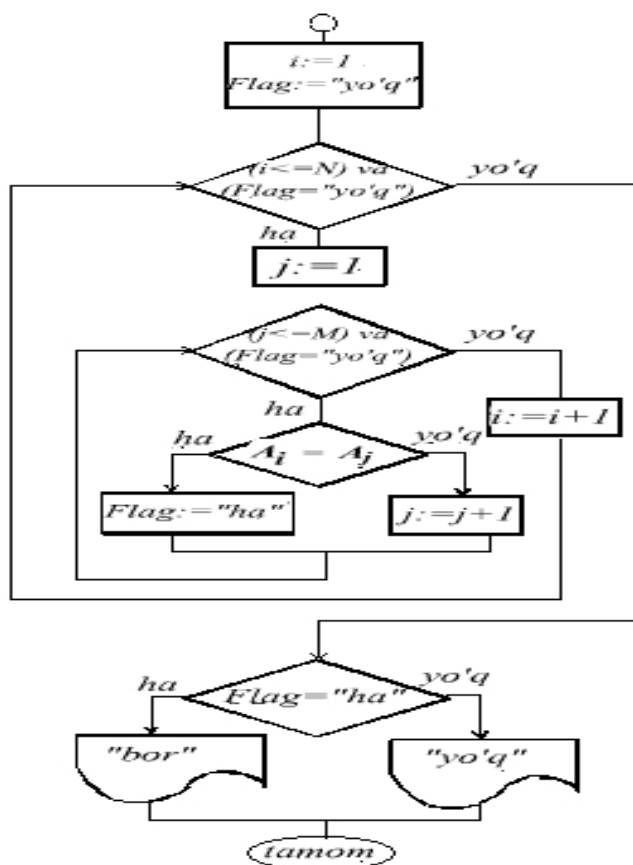
boshl but i, j, lit Flag

```

i:=1; Flag:="Yo'q"
sb toki (i<=N-1) va (Flag="Yo'q")
  j:=i+1
  sb toki (j<=N) va (Flag="Yo'q")
    agar A[i]=A[j]
      u holda Flag:="Ha"
    aks holda j:=j+1
  hal bo'ldi
so
i:=i+1
so
agar Flag="Xa"
  u holda S:=" Mos tushuvchi son bor "
  aks holda S:=" Mos tushuvchi son yo'q "
hal bo'ldi
tamom

```

blok sxemasi:



Algoritmnining bajarilishi

Tekshirilayotgan shartning belgilanishi:

$(i \leq N-1) \text{ va } (\text{Flag} = \text{"Yo'q"}) \Rightarrow (1)$

$(i \leq N) \text{ va } (\text{Flag} = \text{"Yo'q"}) \Rightarrow (2)$

N test	i	Flag	(1)	j	(2)	A[i]=A[j]	S
--------	---	------	-----	---	-----	-----------	---



1	1	"Yo'q"	+	2	+	-	
				3	+	-	
				4	+	-	
	5			- (so)			
1	2	"Ha"	+	3	+	-	
				4	+	+	
				- (so)			
	3		- (so)				" Mos tushuvchi son bor "
2	1	"Yo'q"	+	2	+	-	
	2		+	3	+	-	
	3		- (so)	4	- (so)	-	
				3	+		
				4	- (so)		
						" Mos tushuvchi son yo'q "	

Turbo Pascaldagi dasturi:

Program Equal;

Type Mas = Array [1..20] of Integer;

Var A : Mas; i, j, N : Integer; Flag : Boolean;

{-----}

Begin

Write('N = '); ReadLn(N);

For i := 1 to N do

begin Write('A[', i, '] = '); ReadLn(A[i]) end;

WriteLn('A Massiv');

For i := 1 to N do Write(A[i] : 4);

WriteLn

{-----}

i:=1; Flag:= FALSE;

While (i<=N-1) and not Flag do

begin

j:=i+1;

While (j<=N) and not Flag do

If A[i]=A[j] then Flag:=TRUE else j:=j+1;

i:=i+1

end;

{-----}

WriteLn('Javob :');

If Flag then WriteLn(' Mos tushuvchi son bor.')

else WriteLn(' Mos tushuvchi son yo' ' q.');

ReadLn

END.

**5.2 - misol.** Berilgan butun qiymatli  $A(N, N)$  massivning bosh diagonal qiy qismida joylashgan elementlar ichida manfiy sonlar mavjudmi aniqlang.

Test				
Test	Tekshirish	Berilgan		Natija
		N	A Massiv	S
1	mavjud	4	1 -1 2 1 2 3 1 0 1 -1 2 -1 -2 1 0 1	"Manfiy son bor"
2	mavjud emas	3	1 -1 2 1 0 1 2 1 1	" Manfiy son yo'q"

Algoritmi:

alg diagonal (but N, but jad  $A[1:N, 1:N]$ , lit S)

arg N,A

natija S

boshl but i, j, lit Flag

Flag:="Yo'q"; i:=2

sb toki ( $i \leq N$ ) va (Flag="Yo'q")

j:=1

sb toki ( $j < i$ ) va (Flag="Yo'q")

agar  $A[i, j] < 0$

u holda Flag:="Ha"

aks holda j:=j+1

hal bo'ldi

so

i:=i+1

so

agar Flag="Ha"

u holda S:=" Manfiy son bor "

aks holda S:=" Manfiy son yo'q "

hal bo'ldi

tamom

Algoritmning bajarilishi

Tekshirilayotgan shartning belgilanishi:

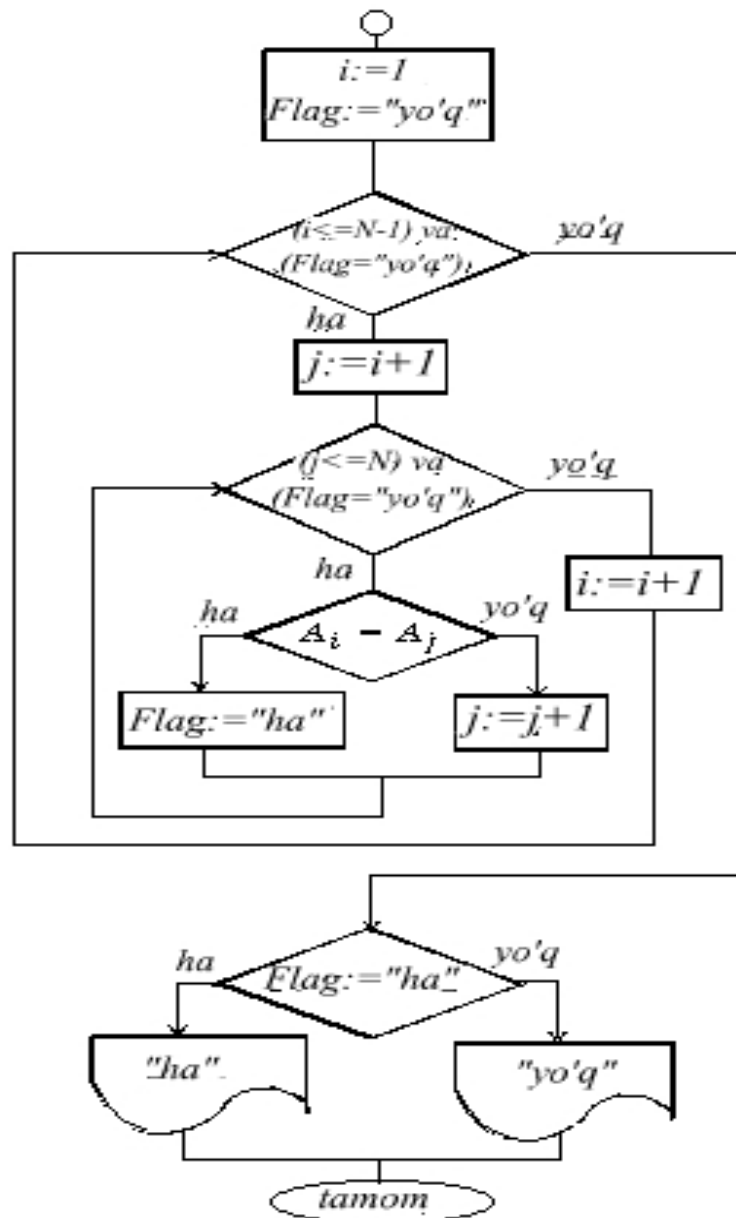
$(i \leq N)$  va (Flag = "Yo'q")  $\Rightarrow$  (1)

$(j < i)$  va (Flag = "Yo'q")  $\Rightarrow$  (2)

N test	i	Flag	(1)	j	(2)	$A[i]=A[j]$	S
1	2	"Yo'q"	+	1	+	-	
				2	-(so)		

	3	"Ha"	+	1 2 3	+ + -(so)	- +	
	4		-(so)				" Manfiy son bor"
2	2	"Yo'q"	+	1 2	+ -(so)	-	
	3		+	1 2 3	+ + -(so)	- -	
	4		-(so)				" Manfiy son yo'q"

blok sxemasi



Turbo Pascaldagi dasturi:

```

Program UnderDiagonal;
  Type Mas = Array [1..10, 1..10] of Integer;
  Var A    : Mas;
      N, i, j : Integer;
      Flag  : Boolean;
  {-----}
  Begin
  ReadLn(N);
  For i := 1 to N do
    For j := 1 to N do
      begin Write('A[', i, ', ', j, ']= ? ');
             ReadLn(A[i, j])
            end; WriteLn;
  WriteLn('Matritsa :');
  For i := 1 to N do
  begin
    For j := 1 to N do Write(A[i, j] : 5);
    WriteLn
  end; WriteLn {-----}
  i := 2 ; Flag := FALSE;
  While (i<=N) and not Flag do
  begin      j:=1;
    While (j<i) and not Flag do
      If (A[i, j]<0)
        then Flag:=TRUE
        else j:=j+1;
    i:=i+1
  end; {-----}
  WriteLn('Javob :');
  Write('Bosh diagonal quyi qismida joylashgan elementlar orasidan');
  If Flag then WriteLn(' Manfiy son bor')
    else WriteLn(' Manfiy son yo' q');
  ReadLn
END.

```

5.3 - Misol. "Spartak" va "Zenit" basketbol jamoasida bir xil bo'yli o'yinchilar bormi tekshiring.

Test

Quyidagi belgilashlarni kiritamiz:

N - "Spartak" jamoasi o'yinchilari soni;

M - "Zenit" jamoasi o'yinchilari soni;

S(N) - "Spartak" jamoasi o'yinchilari bo'ylaridan tashkil topgan massiv (sm);

Z(N) - "Zenit" jamoasi o'yinchilari bo'ylaridan tashkil topgan massiv (sm).

Test	Tekshirish	Berilgan	Natija

		Spartak		Zenit		Satr
		N	S(N)	M	Z(M)	
1	Bor	3	200 195 205	4	198 200 206 192	"bo'ylari bir xil o'yinchilar bor (sm)"
2	Yuk	2	200 195	2	198 201	" bo'ylari bir xil o'yinchilar yo'q (sm)"

*Algoritmi:*

**alg** Bo'yi (**but** N, M, **but jad** S[1:N], Z[1:M], **lit** Satr)

**arg** N,M,S,Z

**natija** Satr

**boshl but** i, j, **lit** Flag

i:=1; Flag:="Yo'q"

**sb toki** (i<=N) **va** (Flag="Yo'q")

j:=1

**sb toki** (j<=M) **va** (Flag="Yo'q")

**agar** S[i]=Z[j]

**u holda** Flag:="Ha"

**aks holda** j:=j+1

**hal bo'ldi**

**so**

i:=i+1

**so**

**agar** Flag="Ha"

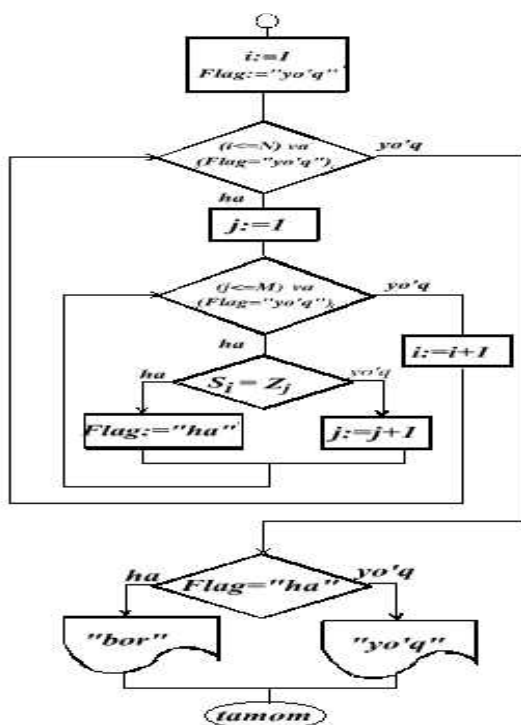
**u holda** Satr:=" bo'ylari bir xil o'yinchilari bor"

**aks holda** Satr:= bo'ylari bir xil o'yinchilari yo'q"

**hal bo'ldi**

**tamom**

blok-sxemasi fragmenti:



*Algoritmnining bajarilishi*

Tekshirilayotgan shartning belgilanishi:

$(i \leq N) \text{ va } (\text{Flag} = \text{"Yo'q"}) \Rightarrow (1)$

$(j < i) \text{ va } (\text{Flag} = \text{"Yo'q"}) \Rightarrow (2)$

test	i	Flag	(1)	j	(2)	S[i]=Z[j]	Satr	
1	1	"Yo'q"	+	1	+	-		
	2	"Ha"	-(so)	2	-	+		
2	1	"Yo'q"	+	1	+	-		
				2	+	-		
				3	-(so)			
	2			+	1	+	-	
					2	+	-	
					3	-(so)		
3			-(so)				"Yo'q"	

Turbo Pascaldagi dasturi:

Program EqualHeight;

Type Mas = Array [1..20] of Integer;

Var

Spart, Zenit : Mas; N, M, i, j : Integer; Flag: Boolean; Name : String;

{-----}

Begin

write('Spartak o'yinchilar sonni N='); ReadLn(N);

For i := 1 to N do

begin Write(i, '-nchi o'yinchi '); ReadLn(Spart[i]) end; WriteLn;

write('Zenit o'yinchilar sonni M='); ReadLn(M);

```

For i := 1 to M do
  begin Write(i, '-nchi o`yinchi '); ReadLn(Zenit[i]) end;
  WriteLn;
{-----}
i:=1; Flag:=FALSE;
While (i<=N) and not Flag do
  begin
  j:=1;
  While (j<=M) and not Flag do
    If Spart[i]=Zenit[j] then Flag:=TRUE else j:=j+1;
  i:=i+1
  end;
{-----}
  Write('Javob: Spartak va Zenit komandalarida bir xil bo`yli');
  If Flag then Write(' o`yinchilar bor ') else Write(' o`yinchilar yo`q ');
  ReadLn
END.

```

**5.4 - misol.** *Shinalar to'plamidan diametri -D sm, og'irligi-W grammdan ko'p bo'lmagan farqdagi ikki shinani tanlash.*

#### Test

N	Tekshirish	Berilgan					Natija
		N shinalar	Diametr	Og'irligi	Farq		S
					diam.	Og'ir.	
1	Bunday shinalar bor	1 2 3 4	103 100 99 101	98 100 101 99	1	1	"2 va 3-shinalar"
2	Bunday shinalar yo'q	1 2 3	100 98 100	100 100 98	1	1	" Bunday shinalar yo'q "

#### Algoritmi:

**alg** Shina (**but** N, Shina1, Shina2, **haq jad** Diam[1 : N],  
Og'ir[1 : N], **haq** FarqDiam, FarqOg'ir, **lit** S)  
**arg** N, Diam, Og'ir; **natija** S, Shina1, Shina2  
**boshl but** i, j, **lit** Flag ; i:=1; Flag:="Yo'q" ;  
**sb toki** (i<=N-1) **va** (Flag="Yo'q") j:=i+1  
**sb toki** (j<=N) **va** (Flag="Yo'q")  
**agar** (abs(Diam[i] - Diam[j]) <= FarqDiam) **va** (abs(Og'ir[i] - Og'ir[j]) <= FarqOg'ir )  
**u holda** Flag:="Xa"; Shina1:=i; Shina2:=j

**aks holda** j:=j+1

**hal bo' ldi**

**so**

i:=i+1

**so**

**agar** Flag="Xa" **u holda** S := "Parametrlar bo' yicha mos keluvchi shinalar- " + Shina1 + " va " + Shina2 + "shinalar."

**aks holda** S := "To' plamda bunday shinalar yo' q."

**al bo' ldi**

**tamom**

Algoritmning bajarilishi

Tekshirilayotgan shartning belgilanishi:

(i <= N-1) va (Flag = "Yo'q") => (1)

(i < N) va (Flag = "Yo'q") => (2)

(abs(Diam[i] - Diam[j]) <= FarqDiam) va

(abs(Og'ir[i] - Og'ir[j]) <= FarqOg'ir) => (3)

test	I	Flag	(1)	j	(2)	(3)	Shina 1	Shina 2
1	1	"Yo'q"	+	2	+	-		
				3	+	-		
				4	+	-		
				5	-(so)			
	2	"Ha"	+	3	+	+	2	3
					-(so)			
2	1	"Yo'q"	+	2	+	-		
				3	+	-		
				4	-(so)			
	2		+	3	+	-		
				4	+			
	3		-(so)					

Turbo Pascaldagi dasturi:

Program MyTyres;

Type Mas = Array [1..100] of Real;

Var

Number, i, j, First, Second : Integer;

Diameter, Weight : Mas;

Flag : Boolean;

D, W : Real; {-----}

Procedure InputOutput;

Begin



```

ReadLn(Number);
For i := 1 to Number do
  begin
    ReadLn(Diameter[i]);
    ReadLn(Weight[i])
  end;
ReadLn(D); ReadLn(W);
For i := 1 to Number do
  WriteLn(i:4, Diameter[i]:10:1, Weight[i]:10:1);
WriteLn
End; { of InputOutput } {-----}
Procedure YesNo(Var First, Second : Integer; Var Flag : Boolean);
Begin
  i:=1; Flag := FALSE;
  While (i<=Number-1) and not Flag do
    begin
      j := i+1;
      While (j<=Number) and not Flag do
        If (Abs(Diameter[i]-Diameter[j]) <= D) and (Abs(Weight[i]-Weight[j]) <= W)
          then begin Flag:=TRUE; First:=i; Second:=j end
          else j := j+1;
      i:=i+1
    end;
End; {of YesNo } {-----}
BEGIN
  InputOutput;
  YesNo(First, Second, Flag);
  WriteLn('Javob :');
  If Flag then WriteLn(' Parametrlar bo' yicha mos keluvchi shinalar', First, ' va ',
Second, ' shinalar.')
    else WriteLn(' To' plamda bunday shinalar yo' q ');
  ReadLn
END.

```

#### *Mustaqil ishlash uchun masalalar*

**5.1.** Butun qiymatli  $A(N, M)$  matritsadagi berilgan  $K$  songa karrali bo'lgan birinchi musbat elementning indekslarini, agar matritsada bunday element bo'lmasa, shu xaqida xabarni ekranga chop eting.

**5.2.** Butun qiymatli  $A(N, M)$  matritsadagi birinchi manfiy elementni shu matritsa elementlarining eng kattasi bilan almashtiring. Agar matritsada manfiy element bo'lmasa, shu haqida xabarni ekranga chop eting.

**5.3.** Berilgan  $A(N, M)$  matritsaning birinchi manfiy elementi joylashgan satrini uchiring.

**5.4.** Berilgan  $A(N, M)$  matritsaning barcha elementlar o'rta arifmetik qiymatidan katta bo'lgan birinchi elementning indekslarini toping.

5.5. Berilgan  $A(N, M)$  matritsada nolgacha teng birinchi elementi joylashgan satr va ustunni uchiring. Hosil bo'lgan matritsani zichlang.

5.6. Tekislikda nuqtalar to'plami berilgan. Bir-biridan uzoqlashganligi berilgan  $D$  masofadan katta bo'lgan nuqtalar juftliklarini toping.

5.7. Butun qiymatli uchta  $A(N)$ ,  $B(M)$  va  $C(L)$  massivlar berilgan. Uchta massivda ham uchraydigan har bitta sonni toping. Bunday son bo'lmasa, shu haqida xabarni ekranga chop eting.

5.8. Bolalar bog'chasida  $N$  ta koptokchalar bor. Har bir koptokchalar diametri va rangi haqida ma'lumotlar mavjud. Aniqlang:

- a) koptokchalar orasida yuzasi  $900 \text{ sm}^2$  teng bo'lgan darchadan o'tmaydiganlari mavjudmi;
- b) koptokchalar orasida bir xil rangli va diametrlilar mavjudmi;

## 6. For (uchun) va While (toki) tipli sikllar kombinatsiyasi yordamida algoritm va dasturlar tuzish.

Toki/Uchun tipli sikl strukturasi:	Uchun/Toki tipli sikl strukturasi:
<b>sb toki</b> <shart> tashqi sikl tanasi ..... <b>sb i uchun</b> A dan B gacha ichki sikl tanasi <b>so</b> ..... <b>so</b>	<b>sb i uchun</b> A dan B gacha tashqi sikl tanasi ..... <b>sb toki</b> <shart> ichki sikl tanasi <b>so</b> ..... <b>so</b>

6.1-misol. Berilgan butun qiymatli  $A(N, M)$  matritsada nollar mavjud bo'lgan satrlar sonini aniqlang

Berilgan			Natija
N	M	A Matritsa	K
3	3	$\begin{pmatrix} 1 & 0 & 1 \\ 2 & 2 & 1 \\ 0 & 1 & 0 \end{pmatrix}$	2

### Algoritmi

**alg** nul\_satr (**but** N, M, K, **but jad** A[1:N, 1:M])

**arg** N, M, A

**natija** K

**boshl** **but** i, j, **lit** Flag

K := 0

**sb i uchun** 1 dan N gacha

j := 1; Flag := "Yo'q"

**sb toki** (j <= M) va (Flag = "Yo'q")

agar A[i, j] = 0 **u holda** Flag := "Ha"; K := K+1

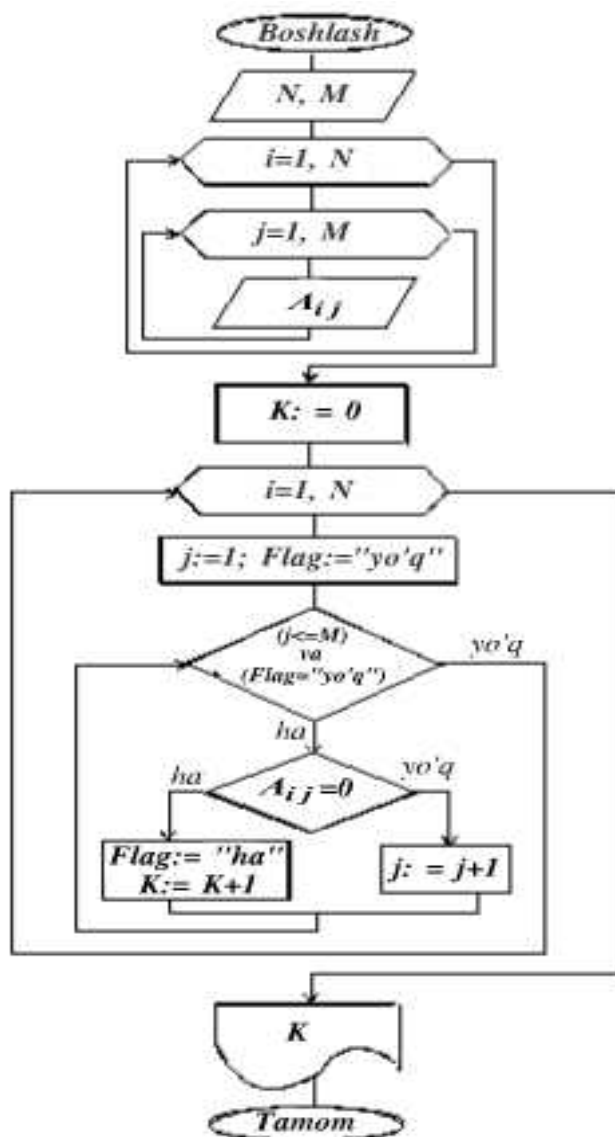
aks holda j := j+1

hal bo'ldi

**so**

**so**

**tamom**  
blok sxemasi:



Algoritmnin bajarilishi

Tekshirilayotgan shartning belgilanishi:

$$(j \leq M) \text{ va } (\text{Flag} = \text{"Yo'q"}) \Rightarrow (1)$$

I	Flag	j	(1)	A[i,j]=0	K
1	"Yo'q"	1	+	-	0
	"Ha"	2	+	+	
2	"Yo'q"	1	+	-	
		2	+	-	
		3	+	-	
		4	-(so)		
3	"Yo'q"	1	+	+	2
	"Ha"		-(so)		

Turbo Pascaldagi dasturi

```

Program ContainZero;
  Var A      : Array[1..10, 1..10] of Integer;
      N, M, i, j, K : Integer; {-----}
Procedure InputOutput;
Begin
  ReadLn(N,M);
  For i := 1 to N do
    For j := 1 to M do
      begin Write('A[', i, ', ', ' ', j, ']= ? ');
            ReadLn(A[i, j])
      end;
  For i := 1 to N do
    begin
      For j := 1 to M do Write(A[i, j] : 5);
      WriteLn
    end;
End; { of InputOutput } {-----}
Function Zero(i:Integer):Boolean;
  Var Flag : Boolean;
Begin
  j:=1; Flag:=FALSE;
  While (j<=M) and not Flag do
    If A[i, j]=0 then Flag:=TRUE else j:=j+1;
  Zero:=Flag
End; {-----}
BEGIN
  InputOutput; K:=0;
  For i := 1 to N do
    If Zero(i) then K:=K+1;
  WriteLn(K); ReadLn
END.

```

**6.2 - misol.** Butun qiymatli  $A(N, M)$  matritsa ustunlari elementlarining eng kattalari ichida berilgan  $K$  butun songa tenglari uchraydimi aniqlang.

**Test**

Test	Tekshirish	Berilgan				Natija
		K	N	M	A matritsa	
1	Uchraydi	5	3	3	$\begin{pmatrix} 1 & 5 & 8 \\ 4 & 2 & 0 \\ 1 & 2 & 3 \end{pmatrix}$	" Uchraydi "
2	Uchramaydi	1	2	2	$\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$	" Uchramaydi "

Algoritmi

alg Ha\_Yo'q (but N,M,K, but jad A[1:N, 1:M], lit S)

arg N,M,K,A; natija S;

boshl but i, j, JMax, lit Flag

Flag:="Yo'q"; j:=1

sb toki (j<=M) va (Flag="Yo'q") JMax:=A[1,j]

sb i uchun 2 dan N gacha

agar A[i,j]>JMax

u holda JMax:=A[i, j]

hal bo'ldi

so

agar K=JMax

u holda Flag:="Xa"

aks holda j:=j+1

hal bo'ldi

so

agar Flag="Xa"

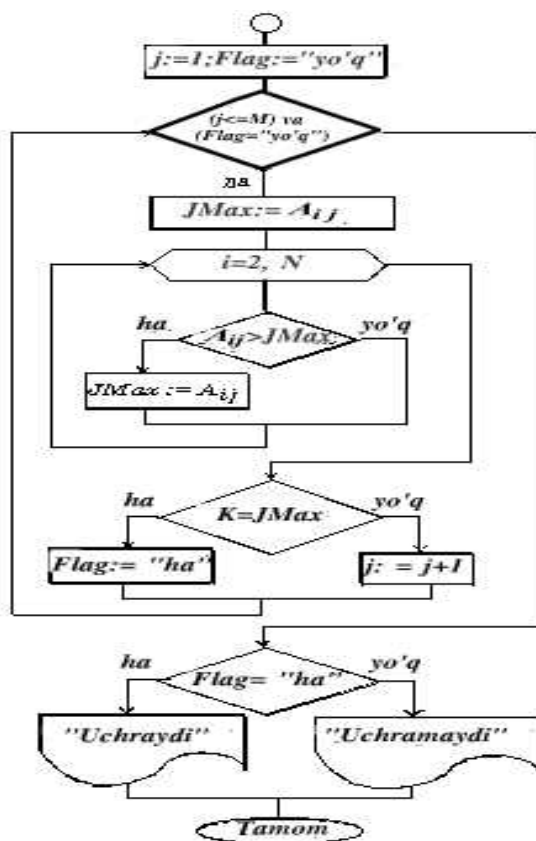
u holda S := " Uchraydi "

aks holda S := " Uchramaydi "

hal bo'ldi

tamom

Blok-sxemasi fragmenti:



## Algoritmning bajarilishi

Tekshirilayotgan shartning belgilanishi:

$(j \leq M)$  va  $(\text{Flag} = \text{"Yo'q"}) \Rightarrow (1)$

test	Flag	J	(1)	Jmax	I	$A[i,j] > Jmax$	$K = Jmax$
1	"Yo'q"	1	+	1	2	+	-
				4	3	-	
2	"Ha"	2	+	5	2	-	+
			-(so)		3	-	
		3	-(so)	2			
2	"Yo'q"	1	+	2	2	-	-
		2	+	1	2	+	-
		3	-(so)	2			

### Turbo Pascaldagi dasturi

Program Checking;

Var A : Array[1..10, 1..10] of Integer;

N, M, i, j, K, JMax: Integer;

Flag : Boolean;

{-----}

Procedure InputOutput;

Begin

ReadLn(K, N, M);

For i := 1 to N do

For j := 1 to M do

begin Write('A[', i, ', ', j, ' ] = ');

ReadLn(A[i, j])

end;

For i := 1 to N do

begin

For j := 1 to M do Write(A[i, j] : 4);

WriteLn

end;

End; { of InputOutput }

{-----}

Procedure YesOrNot(Var Flag:Boolean);

Begin

Flag:=FALSE; j:=1;

While (j<=M) and not Flag do

begin JMax:=A[1, j];

For i := 2 to N do

If A[i, j]>JMax then JMax:=A[i, j];

If K=JMax then Flag:=TRUE else j:=j+1

end;

End;

{-----}

BEGIN

```

InputOutput;
YesOrNot(Flag);
Write('Javob :', K );
WriteLn(' matritsa ustunlarining maksimal elementlar ichida');
If Flag then Write(' uchraydi')
    else Write(' uchraydi ');
    ReadLn
END.

```

**6.3 - misol.** Butun qiymatli  $A(N, M)$  matritsa berilgan. Agar matritsa satrining hech bo'lmaganda biror elementi manfiy bo'lsa, u holda bu satrning barcha elementlarini nollar bilan almashtiring

Test		
Berilgan		Natija
N	A matritsa	A matritsa
3	$\begin{pmatrix} 1 & -2 & 1 \\ 1 & 2 & 1 \\ -1 & 2 & -2 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 0 \\ 1 & 2 & 1 \\ 0 & 0 & 0 \end{pmatrix}$

#### Algoritmi

```

alg Modifikasiya(but N, haq jad A[1:N, 1:N])
  boshl but i, j, lit Flag
  kiritish N
  sb i uchun 1 dan N gacha
    sb j uchun 1 dan N gacha
      kiritish A[i,j]
      so
      sb i uchun 1 dan N gacha
        j := 1; Flag := "Yuk"
        sb toki (j<=N) va (Flag = "Yo' q")
          agar A[i, j]<0      u holda Flag := "Ha"
            aks holda j:=j+1
          hal bo' ldi
        so
        agar Flag = "Ha"  u holda
          sb j uchun 1 dan N gacha A[i, j]:=0
          so
        hal bo' ldi
      so
    sb i uchun 1 dan N gacha
      sb j uchun 1 dan N gacha

```

chiqarish  $A[i,j]$

so

so

tamom

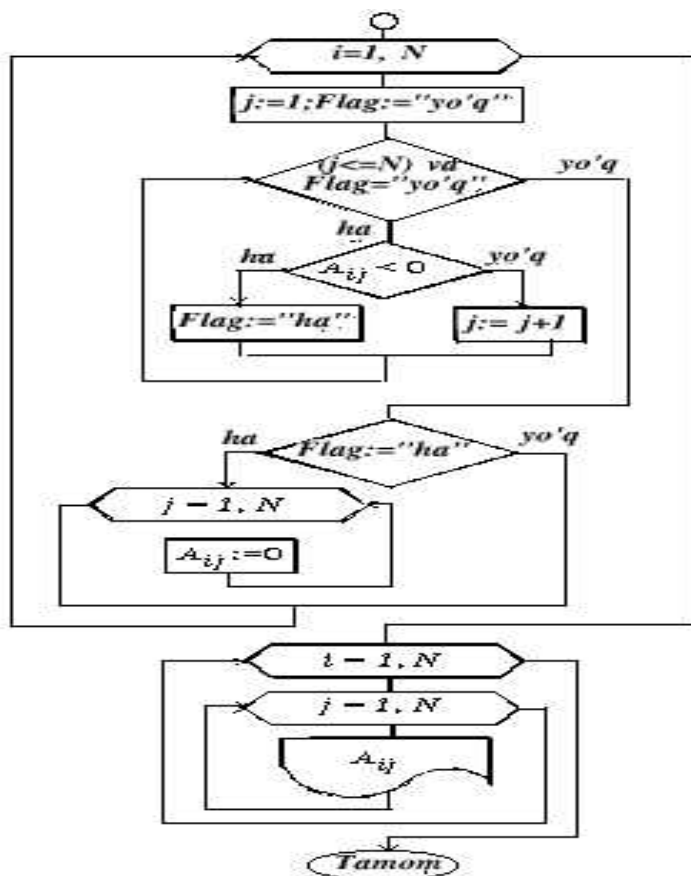
*Algoritmning bajarilishi*

Tekshirilayotgan shartning belgilanishi:

$(j \leq N)$  va  $(\text{Flag} = \text{"Yo'q"}) \Rightarrow (1)$

i	Flag	j	(1)	$A[i,j] < 0$	Flag="Ha"	$A[i,j]$
1	"Yo'q" "Ha"	1	+	-	+	$A[1,1]=0$ $A[1,2]=0$ $A[1,3]=0$
		2	+	+		
		1	-(so)			
		2				
2	"Yo'q"	1	+	-	-	
		2	+	-		
		3	+	-		
		4	-(so)			
3	"Yo'q" "Ha"	1	+	+	+	$A[3,1]=0$ $A[3,2]=0$ $A[3,3]=0$
		1	-(so)			
		2				
		3				

*blok-sxemasi fragmenti:*



*Turbo Pascaldagi dasturi:*

Program Modify;



```

Var A      : Array[1..10, 1..10] of Real;
    N, i, j : Integer;
Procedure InputOutput;
Begin
    ReadLn(N);
    For i := 1 to N do
        For j := 1 to N do
            begin Write(' A[', i, ', ', j, ' ] = ');
                    ReadLn(A[i, j])
            end;
        For i := 1 to N do
            begin
                For j := 1 to N do Write(A[i, j] : 5 : 1);
                WriteLn
            end;
        End; { of InputOutput }
    {-----}
Procedure Line(Var i : Integer);
    Var Flag : Boolean;
    Begin
        j := 1; Flag := FALSE;
        While (j<=N) and not Flag do
            If A[i, j]<0 then Flag:=TRUE else j:=j+1;
        If Flag then
            For j := 1 to N do A[i, j] := 0
    End;
    {-----}
Procedure OutRes;
    Begin
        WriteLn(' Natija- Matritsa: '); WriteLn;
        For i := 1 to N do
            begin
                For j := 1 to N do Write(A[i, j]:5:1);
                WriteLn
            end; ReadLn
    End; { of OutRes }
BEGIN
    InputOutput;
    For i := 1 to N do Line(i);
    OutRes;      END.

```

*Mustaqil ishlash uchun masalalar*

- 6.1.**  $A(N, N)$  matritsa berilgan.  $V$  o'zgaruvchiga  $A$  marisadagi hech bo'lmaganda bitta nol element bo'lgan satrlar sonni ta'minlang.
- 6.2.** Berilgan  $A(N, M)$  matritsada manfiy element bo'lmagan satrlar sonini aniqlang.
- 6.3.** Bir o'lchovli massivdagi har uchinchi musbat elementni o'chiring.
- 6.4.**  $A(N, N)$  matritsaning har bir satridagi eng katta tub sonni aniqlang. Agar satrda tub son bo'lmasa mos xabarni chop eting.
- 6.5.** Mukammal son deb, o'zining bo'luvchilari yig'indisiga teng songa aytiladi. Masalan, 28 mukammal son, chunki  $1+2+3+4+7+14=28$ .  $[1,100]$  oraliqdagi barcha mukammal sonni toping.
- 6.6.** Pifagor sonlari deb,  $a^2 + b^2 = c^2$  tenglamani qanoatlantiruvchi  $a, b, s$  natural sonlar uchligiga aytiladi. Masalan, 6, 8, 10 sonlar uchligi pifagor sonlari hisoblanadi. 25 dan oshmaydigan barcha pifagor sonlarini toping.
- 6.7.**  $N \times M$  tartibli matritsa berilgan. Shunday  $B$  massiv tuzingki, agar matritsaning  $k$ -ustun elementlari nol bo'lsa, uning  $k$ -elementiga 0, aks holda 1 qiymat bering.
- 6.8.**  $N \times M$  tartibli matritsa berilgan. Shunday  $B$  massiv tuzingki bunda agar matritsaning  $k$ -ustun elementlari kamayish bo'yicha tartiblangan bo'lsa, uning  $k$ -elementiga 1, aks holda 0 qiymat bering.
- 6.9.**  $N \times M$  tartibli matritsa berilgan. Shunday  $B$  massiv tuzingki bunda agar matritsaning  $k$ -ustun elementlari simmetrik bo'lsa, uning  $k$ -elementiga 1, aks holda 0 qiymat bering.
- 6.10.**  $N \times M$  tartibli matritsa berilgan. Matritsaning «maxsus» elementlari soni  $k$  – ni aniqlang, «maxsus» element hisoblanadi, agar u o'z usunidagi boshqa qolgan elementlari yig'indisidan katta bo'lsa.

## ADABIYOTLAR

1. Абрамов С.А. и др. Задачи по программированию.-М.:Наука, 1988.-224 стр.
2. Ахо А., Хопкрофт Дж. Построение и анализ вычислительных алгоритмов. - М: Мир, 1979 г., 535 с.
3. Вирт Н.. Алгоритмы и структуры данных. – Досса, Хамарайан, 1997.
4. Кнут Д. Искусство программирования для ЭВМ. Основные алгоритмы.-М: Мир, 2000 г.
5. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ. М.: МЦНМО, 2001.- 960 с.

6. Лебедев В.И. Введение в системы программирования. М: Статистика, 1975.
7. Поляков Д.Б., Круглов И.Ю. Программирование в среде Turbo Pascal: Справ.-метод. пособие.- М.: Изд-во МАИ, 1992.-576 с.
8. Попов В.В. Общение с ЭВМ на естественном языке. М:Наука, 1982.
9. Тыгуу Х. Концептуальное программирование. М: Наука, 1984.
10. Успенский В.А., Семенов А.Л.. Теория алгоритмов: основные открытия и приложения. – М: Наука, 1987, 287 с.
11. Файсман А. Профессиональное программирование на Турбо-Паскале.-Info&F, 1992.-270 стр.

## 1. MUSTAQIL ISHLAR TIZIMI

№	<i>Mavzu nomi</i>	So- at	Topshiriklar	Xisobot shakli
1	2	3		4
1	Algoritmlar nazariyasi tarixini o'rganish.	1	Algoritmlar nazariyasi tarixini, maqsad va vazifalarini yoritib berish	Yozma
2	Birinchi algoritmlarni tuzishga misollar.	1	Matematik misollarni ehadigan algoritmlarni tuzishga misollar ko'rsatish	Yozma
3	Algoritmlar sifatini baholashning asosiy mezonlari.	1	Algoritmlar sifatini baholashning asosiy	Yozma

			mezonlarini o'rganish, ular xarakteristikalarini ko'rsatib berish	
4	Matematik induksiya usulini o'rganish.	1	Matematik induksiya usuli asosida sbotlash usullarga misol ko'rsatish	Yozma
5	Butun qiymatli funksiyalar.	1	Butun qiymatli funksiyalarni o'rganish .	Yozma
6	Binomial koeffitsiyentlar.	1	Binomial koeffitsiyentli funksiyalarga misollar ko'rsatish.	Yozma
7	Algoritm tarifini va xususiyatlarini o'rganish.	1	Algoritm xususiyatlari haqida batafsil ma'lumot berish	Yozma
8	Masala quyilishiga misollar.	1	Masala quyilishida ifodalovchi va o'zgaruvchilarni aniqlashni o'rganish .	Yozma
9	Modellarni qurishga misollar.	1	Modellarni qurishga misollar.	Yozma
10	Algoritmning tug'riligini tekshirishga misollar.	1	Algoritmning tug'riligini tekshirish uchun misollar tuzishni o'rganish	Yozma
11	Xujjatlashtirishga misollar.	1	Yaratilgan algoritm va dasturni izohlashni o'rgaish	Yozma
12	Algoritmning umumiy ko'rinishiga misollar	1	Chiziqli jarayonlarni ifodalovchi algoritmlarga misollar ko'rsatish	Yozma
13	Tarmoqlanish buyruqlariga misollar.	1	Tarmoqlanuvchi jarayonlarni ifodalovchi algoritmlarga misollar ko'rsatish	Yozma
14	Tanlash buyruqlariga misollar.	1	Tanlash jarayonlarni ifodalovchi algoritmlarga misollar ko'rsatish	Yozma
15	Takrorlanish buyruqlariga misollar.	1	Takrorlanuvchi jarayonlarni ifodalovchi algoritmlarga misollar ko'rsatish	Yozma
16	Maksimum va minimum topish algoritmlarini o'rganish.	1	Maksimum va minimum topish masalalarini ehadigan algoritmlarini tuzishni o'rganish.	Yozma
17	EKUB va EKUKlarni topish kabi masalalar algoritmlarini o'rganish.	1	EKUB va EKUKlarni topish masalalarini ehadigan algoritmlarini tuzishni o'rganish.	Yozma
18	Tasvirlarni tanish masalalariga algoritmlarni o'rganish.	1	Tasvirlarni tanish masalalariga algoritmlarni o'rganish.	Yozma
19	Evristik algoritmlarini xususiyatlarini o'rganish.	1	Evristik algoritmlarni tuzishni o'rganish.	Yozma
20	Kommivoyajer masalalari.	1	Kommivoyajer masalalar turlarinini o'rganish.	Yozma

21	Qirralar va chegaralar usuli yerdamida yechiladigan masalalar.	1	Qirralar va chegaralar usuli yerdamida yechiladigan masalalar.	Yozma
22	Eng qisqa yo'larni topish masalalariga algoritmlar tuzish.	1	Eng qisqa yo'larni topish masalalariga algoritmlarni ko'rsatish.	Yozma
23	Tartiblash usullari turlari.	1	Tartiblash usullari turlari.	Yozma
24	Tartiblash masalalarini yechishda rekursiv va rekursiv bo'lmagan algoritmlardan foydalanish.	1	Tartiblash masalalarini yechishda rekursiv va rekursiv bo'lmagan algoritmlardan foydalanishni o'rganish.	Yozma
25	Matrisalarni ko'paytirish masalasiga algoritmlar.	1	Matrisalarni ko'paytirish uchun algoritmlarni tuzishni o'rganish.	Yozma
26	Graflarni amalga oshirish algoritmlari.	1	Graflarni amalga oshirish algoritmlar bilan ishlashni o'rganish.	Yozma
27	Geometrik algoritmlar.	1	Geometrik algoritmlarni tahlil qilishni o'rganish.	Yozma
28	To'rlar va daraxtlar. Daraxtlar tasniflanishi.	1	To'rlar va daraxtlar nazariyasi haqida ma'lumotga ega bo'lish	Yozma
29	Daraxtlar bilan ishlash algoritmlari	1	Ikkilik daraxtlar bilan amallar, daraxtlarda izlash va ma'lumotlarni qushish va boshqalar	Yozma
30	NP-to'liqlik.	1	NP-to'liqlik haqida ma'lumotga ega bo'lish	Yozma
31	Algoritmning hisoblash murakkabligini tushunchasi.	1	Algoritmning hisoblash murakkabligini aniqlashni o'rganish .	Yozma
32	Algoritmni dastur sifatiga ta'sirini hisobli taxlili.	1	Algoritmni dastur sifatiga amalgam oshirishda hisobli taxlilni o'rganish.	Yozma
	Jami	32		

## 7. ORALIQ VA YaKUNIY NAZORAT SAVOLLARI

«Aloritmlar nazariyasi» fani. 2010-2011 o'quv yili 1-semestr.

1. Aloritmlar nazariyasi tarixini o'rganish.
2. Birinchi algoritmlarni tuzishga misollar.
3. Aloritmlar sifatini baholashning asosiy mezonlari.
4. Matematik induksiya usulini o'rganish.
5. Butun qiymatli funksiyalar.
6. Binomial koeffitsiyentlar.
7. Algoritm tarifini va xususiyatlarini o'rganish.

8. Masala quyilishiga misollar.
9. Modellarni qurishga misollar.
10. Algoritmning tug'riligini tekshirishga misollar.
11. Xujjatlashtirishga misollar.
12. Algoritmning umumiy ko'rinishiga misollar
13. Tarmoqlanish buyruqlariga misollar.
14. Tanlash buyruqlariga misollar.
15. Takrorlanish buyruqlariga misollar.
16. Maksimum va minimum topish algoritmlarini o'rganish.
17. EKUB va EKUKlarni topish kabi masalalar algoritmlarini o'rganish.
18. Tasvirlarni tanish masalalariga algoritmlarni o'rganish.
19. Evristik algoritmlarini xususiyatlarini o'rganish.
20. Kommivoyajer masalalari.
21. Qirralar va chegaralar usuli yerdamida yechiladigan masalalar.
22. Eng qisqa yo'larni topish masalalariga algoritmlar tuzish.
23. Tartiblash usullari turlari.
24. Tartiblash masalalarini yechishda rekursiv va rekursiv bo'lmagan algoritmlardan foydalanish.
25. Matrisalarni ko'paytirish masalasiga algoritmlar.
26. Graflarni amalga oshirish algoritmlari.
27. Geometrik algoritmlar.
28. To'rlar va daraxtlar. Daraxtlar tasniflanishi.
29. Daraxtlar bilan ishlash algoritmlari
30. NP-to'liqlik.
31. Algoritmning hisoblash murakkabligini tushunchasi.
32. Algoritmni dastur sifatiga ta'sirini hisobli taxlili.

## **9. «AXBOROT XAVFSIZLIGI» FANI BUYICHA ORALIK VA YAKUNIY NAZORAT VARIANTLARI(BILETLAR).**

### **9.1 ORALIQ NAZORAT BILETLARI**

#### **1 –оралиқ назорати**

##### Variant 1

1. Algoritmash fani va algoritmash san'ati.
2. Algoritmni to'liq tuzishning asosiy bosqichlari. Algoritmni to'g'riligini tekshirish. Algoritmni amalga oshirish.
3. Algoritmni va uning murakkabligini taxlil qilish.

##### Variant 2

1. Algoritmning matematik asoslari. Matematik induksiya. Yig'indi va ko'paytmalar.
2. Algoritmning asosiy ta'rifi va hossalari. Algoritmni tuzishning asosiy bosqichlari. Masala quyilishi.
3. Algoritm murakkabligini baholash uchun mavjud mezonlar. Algoritm murakkabligini vaqt murakkabligi mezonini bo'yicha taxlil qilish.

#### Variant 3

1. Algoritmning matematik asoslari. Butun qiymatli funksiyalar. O'rin almashtirishlar va faktoriallar.
2. Algoritmni tuzishning asosiy bosqichlari. Modelni yaratish. Algoritmni ishlab chiqish.
3. Algoritmni tavsiflash haqida kelishuv. «Maktab algoritmik tili»da algoritmning umumiy ko'rinishi va asosiy buyruqlari.

#### Variant 4

1. Algoritmning matematik asoslari. Binomial koeffitsiyentlar. Fibonachchi sonlari.
2. Algoritmni to'liq tuzishning asosiy bosqichlari. Programmani tekshirish va xujjatlashtirish.
3. Algoritmni ishlab chiqish usullari.

### **2-oralik variantlari**

#### Variant 1

1. Maksimum topish algoritmini tadqiq qilish.
2. Kommivoyajer masalasini yechish uchun GTS algoritmi. Evristik algoritmlar.
3. Deykstra algoritmining psevdokoddagi ko'rinishi va uning baxosi.

#### Variant 2

1. Yevklid algoritmini takomillashtirish.
2. Kommivoyajer masalasi yechimlarini tarmoqlanish va chegaralar usuli bilan tadqiq qilish.
3. Tartiblash usullari. Xoara algoritmi.

### Variant 3

1. Tasvirlarni tanish masalasining algoritmini asimptotik baxosini yaxshilash.
2. Eng qisqa yo'llar. Deykstra algoritmining so'zli tavsifi va chizmasi.
3. Matrisalarni ko'paytirish uchun Shtrassen algoritmi.

### 9.2. YAKUNIY NAZORAT BILETLARI

**O'zbekiston Respublikasi Oliy va O'rta maxsus ta'lim vazirligi  
Alisher Navoiy nomidagi Samarqand davlat universiteti**

Fakultet: Mexanika-matematika

Yo'nalish: 5480100 –

amaliy matematika va informatika

O'quv yili: 2010-2011

Kurs: 2

Semestr: 3

Fan: «Algoritmlar nazariyasi»

Variant 1

1. Algoritmash fani va algoritmash san'ati.
2. Maksimum topish algoritmini tadqiq qilish.
3. Natural  $n$  va  $a_1, \dots, a_n$  xaqiqiy sonlar berilgan.  
 $\text{Min}(a_2, a_4, \dots) + \text{Max}(a_1, a_3, \dots)$ ni xisoblaydigan algoritm va dastur tuzing.

Kafedra mudiri:

prof. I.I. Jumanov

Variant 2



1. Algoritmning matematik asoslari. Matematik induksiya.
2. Kommivoyajer masalasini yechish uchun GTS algoritmi. Evristik algoritmlar.
3. Natural  $n$ , haqiqiy  $n \times 9$  ulchamli matrisa berilgan. Har bir ustunning o'rtta arifmetigini topadigan algoritm tuzing

#### Variant 3

1. Algoritmning matematik asoslari. Yig'indi va ko'paytmalar. Butun qiymatli funksiyalar.
2. Deykstra algoritmining psevdokoddagi ko'rinishi va uning baxosi.
3.  $x_1=y_1=1$ ;  $x_2=y_2=2$ ;  $x_i = \frac{y_{i-1} - y_{i-2}}{i}$ ;  $y_i = \frac{x_{i-1}^2 + x_{i-2} + y_{i-1}}{i!}$ ,  $i=3,4,\dots$  berilgan.  $x_1, y_1, x_2, y_2, \dots, x_{25}, y_{25}$  ni hisoblaydigan algoritm tuzing

#### Variant 4

1. Algoritmning matematik asoslari. O'rin almashtirishlar va faktoriallar.
2. Yevklid algoritmini takomillashtirish.
3.  $n \times m$  o'lchamli matrisa berilgan. Har bir satrni yig'indisini topadigan algoritm tuzing

#### Variant 5

1. Algoritmning matematik asoslari. Binomial koeffitsiyentlar. Fibonachchi sonlari.
2. Kommivoyajer masalasi yechimlarini tarmoqlanish va chegaralar usuli bilan tadqiq qilish.
3.  $n \times m$  o'lchamli matrisa berilgan. Matrisani eng kichik elementi va indeksini aniqlaydigan algoritm tuzing

#### Variant 6

1. Algoritmning asosiy ta'rifi va hossalari.
2. Tartiblash usullari. Xoara algoritmi.
3.  $n = \{5;10;15\}$  kiymatlar uchun  $F = n!$  ni prosedura yordamida hisoblaydigan algoritm tuzing

#### Variant 7

1. Algoritmni to'liq tuzishning asosiy bosqichlari. Masala quyilishi.
2. Tasvirlarni tanish masalasining algoritmini asimptotik bahosini yaxshilash.
3.  $x_1, x_2, \dots$  ketma-ketlik  $x_1=1$ ;  $x_2=0.3$ ;  $x_i=(i+1) \cdot x_{i-2}$ ,  $i=3,4,\dots$  qonuniyat bilan aniqlangan.  $x_1, x_2, \dots, x_{20}$  ni hosil qiladigan algoritm tuzing

#### Variant 8

1. Algoritmni to'liq tuzishning asosiy bosqichlari. Modelni yaratish. Algoritmni ishlab chiqish.
2. Eng qisqa yo'llar. Deykstra algoritmining so'zli tavsifi va chizmasi.
3.  $n = \{5;10;15\}$  kiymatlar uchun  $F = \sum_{i=1}^n i!$  ni prosedura yordamida hisoblaydigan algoritm tuzing

#### Variant 9

1. Algoritmni to'liq tuzishning asosiy bosqichlari. Algoritmni to'g'riligini tekshirish.
2. Matrisalarni ko'paytirish uchun Shtrassen algoritmi.

3.  $n \times m$  o'lchamli matrisa berilgan. Har bir ustunni ko'paytmasini topadigan algoritm tuzing

#### Variant 10

1. Algoritmni to'liq tuzishning asosiy bosqichlari. Algoritmni amalga oshirish.
2. Maksimum topish algoritmini tadqiq qilish.
3.  $n = \{2; 5; 8\}$  qiymatlar uchun  $Z = \sum_{k=1}^n \frac{x^{2k}}{(2k)}$  ni prosedura yordamida hisoblaydigan algoritm tuzing

#### Variant 11

1. Algoritmni va uning murakkabligini taxlil qilish.
2. Algoritmni ishlab chiqish uslublari
3.  $N$  tartibli kvadrat matrisa berilgan. Hamma elementlari nollardan iborat ustun nomerini aniqlaydigan algoritm tuzing

#### Variant 12

1. Algoritmni to'liq tuzishning asosiy bosqichlari. Programmani tekshirish. Xujjatlashtirish.
2. Algoritmning asosiy boshqaruvchi konstruksiyalarini tasvirlash.
3.  $x \in \{0; 0,1; 0,2; \dots; 1,0\}$   $m = 300; l = 1$  qiymatlar uchun  $U(x) = \sum_{n=1}^m \cos \frac{\pi n}{l} x * \sin \frac{\pi n}{l} x$  ni funksiya yordamida hisoblaydigan algoritm tuzing

#### Variant 13

1. Algoritm fani va san'ati.
2. Kommivoyajer masalasini yechish uchun GTS algoritmi.
3.  $N$  tartibli kvadrat matrisa berilgan. Hamma elementlari bir xil bulgan ustun nomerini aniqlaydigan algoritm tuzing

#### Variant 14

1. Yevklid algoritmini takomillashtirish.
2. Algoritmning matematik asoslari. Matematik induksiya.
3.  $x \in \{0; 0,25; 0,5; 0,75; 1,0\}$   $m = 300; l = 1; a = 1; t \in \{0; 0,5; 1; 1,5; 2\}$  qiymatlar uchun  $U(x, t) = \sum_{n=1}^m \cos \frac{\pi n a}{l} t * \sin \frac{\pi n}{l} x$  ni funksiya yordamida hisoblaydigan algoritm tuzing

#### Variant 15

1. Algoritm qiyinligini baholash uchun mavjud mezonlar.
2. Deykstra algoritmining psevdokoddagi ko'rinishi va uning bahosi.
3. Haqiqiy  $x_1, \dots, x_8$  sonlar berilgan. Quyidagi matrisani hosil qiladigan algoritm tuzing

$$\begin{bmatrix} 1 & 1 & K & 1 \\ x_1 & x_2 & K & x_8 \\ K & K & K & K \\ x_1^7 & x_2^7 & K & x_8^7 \end{bmatrix}$$

#### Variant 16

1. Algoritm murakkabligini vaqtli qiyinlik bo'yicha taxlil qilish.
2. Tasvirlarni tanish masalasining algoritmini asimptotik bahosini yaxshilash.
3.  $x \in \{0;0,25;0,5;0,75;1,0\}$   $m = 300$ ;  $l = 1$ ;  $a = 1$ ;  $t \in \{0;0,5;1;1,5;2\}$  qiymatlar uchun

$$U(x, t) = \sum_{n=1}^m \left( \cos \frac{\pi n a}{l} t * \sin \frac{\pi n a}{l} t \right) * \sin \frac{\pi n x}{l}$$

ni funksiya yordamida hisoblaydigan algoritm tuzing

#### Variant 17

1. Algoritmni ishlab chiqish uslublari.
2. Tartiblash usullari. Xoara algoritmi.
3.  $a = 15$ ,  $n = \{10;20;30\}$  qiymatlar uchun  $y = a \sum_{i=1}^n i^3$  ni prosedura yordamida hisoblaydigan algoritm tuzing

#### Variant 18

1. Maksimum topish masalasi.
2. Algoritmning matematik asoslari. Yig'indi va ko'paytmalar. Butun qiymatli funksiyalar.
3.  $n = \{100;200;300;400\}$  qiymatlar uchun  $z = \frac{1}{n} \sum_{i=1}^n i$  ni prosedura yordamida hisoblaydigan algoritm tuzing

#### Variant 19

1. Yevklid algoritmi.
2. Algoritmning xossalari va to'liq tuzishning bosqichlari
3.  $n = \{5;10;15\}$  qiymatlar uchun  $F = n!$  ni prosedura yordamida hisoblaydigan algoritm tuzing

#### Variant 20

1. Tasvirlarni tanish masalasiga algoritm.
2. Algoritmni to'liq tuzishning masala quyilishi va modelni yaratish bosqichlari.
2. Natural  $n$  va  $a_1, \dots, a_n$  haqiqiy sonlar berilgan.  $b_1, \dots, b_n$  ketma-ketlikni hosil qiling. Bu yerda

$$b_i = \frac{a_i}{1 + (a_1 + \dots + a_i)^2}, i=1, n$$

ni hisoblaydigan algoritm tuzing

#### Variant 21

1. Kommivoyajer masalasi qo'yilishi. Evristik algoritmlar ta'rifi.
2. Algoritmni to'liq tuzishda uning murakkabligini baholash bosqichi.
3.  $x \in \{0;0,25;0,5;0,75;1,0\}$   $m = 300$ ;  $l = 1$ ;  $a = 1$ ;  $t \in \{0;0,5;1;1,5;2\}$  qiymatlar uchun

$$U(x, t) = \sum_{n=1}^m \cos \frac{\pi n a}{l} t * \sin \frac{\pi n}{l} x \quad \text{ni funksiya yordamida hisoblaydigan algoritm tuzing}$$

Variant 22

1. Kommivoyajer masalasini yechishda GTS algoritmi.
2. Algoritmni to'liq tuzishnig algoritmni ishlab chiqish, programmashtirish va hujjatlashtirish bosqichlari.

3.  $n = \{2;5;8\}$  qiymatlar uchun  $Z = \sum_{k=1}^n \frac{x^{2k}}{(2k)}$  ni prosedura yordamida hisoblaydigan algoritm tuzing

Variant 23

1. Shoxlar va chegaralar uslubi bilan masalalarni yechish.
2. Algoritmning asosiy boshqaruvchi konstruksiyalarni tavsiflash.
3. Haqiqiy  $x_1, \dots, x_8$  sonlar berilgan. Quyidagi matrisani hosil qiladigan algoritm tuzing

$$\begin{bmatrix} 1 & 1 & K & 1 \\ x_1 & x_2 & K & x_8 \\ K & K & K & K \\ x_1^7 & x_2^7 & K & x_8^7 \end{bmatrix}$$

Variant 24

1. Eng qisqa yo'llar masala qo'yilishi. Deykstra algoritmining so'zli ko'rinishi.
2. Algoritmni ishlab chiqish uslublari
3. Natural  $n$  va  $a_1, \dots, a_n$  haqiqiy sonlar berilgan.  $b_1, \dots, b_n$  ketma-ketlikni hosil qiling. Bu yerda

$$b_i = \frac{a_i}{1 + (a_1 + \dots + a_i)^2}, \quad i=1, n \text{ ni hisoblaydigan algoritm tuzing}$$

## 10. "ALGORIMLAR NAZARIYASI" FANIDAN TEST TOPSHIRIKLARI

1. Куйидаги бандлардан кайси бирида алгоритм тушунчаси аниқроқ ва туликроқ таърифланган?

А) Алгоритм-куйилган масалани ечиш ёки маълум бир мақсадга эришиш учун ижрочи бажариши зарур булган иш харакатнинг (амалларнинг) тушунарли ва аниқ кетма-кетлигидир.

В) Алгоритм узбек математиги Ал Хоразмий номи билан боглик булиб, унинг европача бузиб айтилишидир.

С) Алгоритм деганда ЭХМ учун тузилган дастурни тушунализ.

D) Алгоритм ижрочига берилган курсатма (йурикнома) булиб хизмат килади.

2. Алгоритм маълум бир ижрочига мулжаллаб тузилади. Агар ижрочи ЭХМ булса, алгоритм кандай ёзилиши керак?

A) Блок схемалар ёрдамида ифодаланиши керак.

B) Сўзлар ёрдамида ёзилиши керак.

C) Сўзлар ва формулалар ёрдамида

D) Жадвал кўринишида ифодаланиши зарур.

3. Алгоритм ва ЭХМ учун дастур тушунчалари орасидаги фарк нимадан иборат?

A) ЭХМга тушунарли тилда ёзилган алгоритм дастурдир.

B) Улар бир хил тушунчалар

C) Хар кандай алгоритм дастур була олади

D) Улар орасида хеч кандай умумийлик йук

4. Алгоритм яратиш жараёнининг боскичларини тугри тартибда жойлаштиринг:

1) Масаланинг куйилиши

2) Алгоритмни ёзиш;

3) Модел тузиш;

4) Алгоритмни амалга ошириш (реализация);

5) Алгоритм тугрилигини текшириш;

6) Дастурни текшириш;

7) Алгоритмни ва унинг мураккаблигини тахлил килиш;

8) Хужжатлаштириш.

A) 1;3;2;5;4;7;6;8.

B) 1;2;3;4;5;6;7;8

C) 2;1;3;4;5;6;8;7

D) 1;4;2;5;3;8;7;6

5. Масаланинг куйилишидан нималар аникланади?

A) Нима берилган ва нимани топиш кераклиги

B) Алгоритмнинг узунлиги

C) Дастурнинг бажарилиш вақти;

D) Зарурий хотира хажми.

6. Куйидаги жумлалардан кайси бири масаланинг математик моделини тузиш жараёнини тугри ифодалайди?

A) Масалани математика тилида тавсифлаш.

B) Масалани блок-схемаларда ифодалаш

C) Алгоритмни тугри танлаш;

D) Масала ечимини топиш;

7. Алгоритмнинг самарадорлигини бахолаш учун мезонлар:

A) хотира хажми ва ижро вақти;

B) аниклик ва тушунарлилик;

C) зарурий хотира хажми;

D) тугрилик ва аниклик

8. Алгоритмни тугри деймиз, агар

A) у куйилган масалага мос ечимни берса;

B) у албатта сонли ечим берса;

C) у охиригача ишласа;

D) у хатолардан холи булса.

9. Алгоритмни аник деймиз, агар

A) Унинг барча кадамлари аник булиб, уларни бошқача талкин қилиш мумкин булмаса;

B) Унинг барча кадамлари сонли натижага олиб келса;

C) Унда математик модел тугри булса;

D) Хотира хажми энг кам микдорда булса.

10. Дастурий таъминотнинг ишончилиги – бу

A) Дастурнинг маълум бир даврда хатоларсиз ишлай олиш хусусияти

B) Дастурнинг тухтовсиз ишлай олиш хусусияти

C) Дастурнинг ихтиёрий турдаги компьютерларга мосланганлиги

D) Дастурнинг узгартиришларга мосланганлиги

11. Дастурий таъминотнинг хусусиятини нима ифодалайди?

A) Дастурий таъминот вазифаларининг (функцияларининг) сони, қуввати ва таъсир доирасининг кенглиги.

B) Дастурий таъминот вазифаларининг аниклиги

C) Унинг ишончилиги;

D) Дастурий таъминот вазифаларининг мақсадларга мослиги

12. Куйидагилардан қайси бирида дастурлаш технологияси тушунчаси тугри тавсифланган?

A) Дастурий маҳсулот яратиш жараёнини утказишнинг самарали усуллари ва воситалари ҳақидаги билимлар мажмуаси.

B) Дастурлаш жараёнини автоматлаштиришга мулжалланган воситалар мажмуаси.

C) Дастурий таъминот яратишга мулжалланган дастурий воситалар ҳақидаги билимлар мажмуаси.

D) Дастурий маҳсулот яратиш жараёнининг илмий тавсифи

13. Модулли структурага эга булган дастур-бу
- A) Кисмий масалаларга мос холда бир неча модуллардан иборат дастур.
  - B) Узаро боғлиқ булмаган бир неча модуллардан иборат дастур.
  - C) Факат бир модулдан иборат дастур
  - D) Процедуралардан фойдаланувчи дастур.
14. Структурали ёзув нимани англатади?
- A) Дастур факат кетма-кетлик, тармокланиш ва такрорлаш конструкцияларидан фойдаланиб тузилган
  - B) Дастур факат кетма-кетлик конструкцияларидан иборат;
  - C) дастур факат кетма-кетлик ва «утиш» конструкцияларидан фойдаланиб тузилган
  - D) дастур факат тармокланиш ва утиш конструкцияларидан фойдаланиб тузилган.
15. Структурали дастурлаш кандай гоёга асосланади?
- A) хар кандай дастурни утиш операторини бир марта хам ишлатмасдан трузиш мумкин.
  - B) хар кандай масала бир неча кисмий масалалардан иборат
  - C) хар кандай дастурлаш тилларида процедура ва функциялар тушунчалари мавжуд.
  - D) хар кандай дастурни кисмий дастурларга булиш мумкин.
16. Структурали дастур хосил килиш учун куйидаги усуллардан кайси бирини ишлатиш мумкин.
- A) куйилаб ёки юкорилаб бориш усулида лойихалаш
  - B) динамик дастурлаш
  - C) кетма-кет лойихалаш
  - D) олдиндан лойихалаш
17. Объектга йуналтирилган дастурлашнинг асосий гоёси?
- A) маълумотлар ва улар устида бажариладиган амалларни бир структурага бирлаштириш;
  - B) маълумотларни объектлар сифатида тавсифлаш;
  - C) маълумотлар ва улар устида бажариладиган амалларни алохида-алохида дастурлаш;
  - D) объектлар тури деган тушунчани киритиш
18. Объектга йуналтирилган дастурлаш куйидаги уч тушунчага асосланади:
- A) инкапсуляция; меросхурлик; полиморфизм
  - B) инкапсуляция; меросхурлик (наследование); статик методлар.
  - C) инкапсуляция; методлар; полиморфизм.
  - D) процедуралар; функциялар; методлар.
19. Объектлар тури нимани ифодалайди?

- A) маълумотлар ва улар устида амаллар бажарадиган процедура, функцияларнинг бирлашмасини.
- B) статик ва динамик методлар бирлашмасини
- C) процедура ва функцияларнинг бирлашмасини
- D) маълумотларнинг узаро бирлашмасини

20. Объектларнинг меросхурлик хусусияти нимани билдиради?

- A) бобо объектда тавсифланган маълумотлар ва методлар меросхур объектга тулик утишини;
- B) бобо объектга тавсифланган барча турлар ва узгарувчилар меросхур объектга тулик утишини;
- C) бобо объектга тавсифланган статик методларнинг меросхур объектга тулик утишини;
- D) бобо объектда тавсифланган динамик методларнинг меросхур объект учун ҳам уринли булишини;

21. Объектнинг компонентлари?

- A) маълумотлар ва процедура, функциялар.
- B) турлар ва процедуралар
- C) узгарувчилар ва нишонлар
- D) операторлар ва маълумотлар

22. Объектнинг нусхаси нима?

- A) объект турига тегишли конкрет узгарувчи;
- B) объект таркибидаги методлар;
- C) объект таркибидаги процедура;
- D) объект турига тегишли конкрет узгармас.

23. Объектлар тури Паскаль–программанинг кайси булимида тавсифланади?

- A) турлар булимида;
- B) нишонлар булимида;
- C) сарлавхада;
- D) процедура ва функциялар булимида;

24. Паскал тилининг процедура ва функциялари таркибида объектлар тавсифланиши мумкинми?

- A) мумкин эмас
- B) мумкин
- C) факат процедураларда мумкин.
- D) факат функцияларда мумкин.

25. Куйидагилардан кайси бири объектнинг компоненти сифатида олиниши мумкин эмас?



- A) файл
- B) процедура
- C) функция
- D) ёзув

26. Куйида

$$y = \frac{(x+1)^2 + 2(x+1)}{2(x+3)^2 + 3(x+3)}$$

ифодани ҳисоблаш учун алгоритмлар келтирилган. Улардан қайси бири энг самарали (эффектив) алгоритм була олади?

A) Бошланиш

y:= x+1

y:=y^2+2y

b:=x+3

b:=2b^2+3b

y:=y/b

Тамом

B) Бошланиш

a:=x+1

b:=x+3

c:=a^2+2a

d:=2b^2+3b

y:=c/d

тамом

C) Бошланиш

a:=x+1

a:=a^2+2a

b:=x+3

b:=2b^2+3b

y:=a/b

Тамом

D) Бошланиш

A:=x+1

$$B:=a^2$$

$$C:=x+3$$

$$D:=c^2$$

$$E:=b+2a$$

$$F:=2d+3c$$

$$Y:=e/f$$

Тамом

27. Куйида икки алгоритм келтирилган:

1-алгоритм: бошланиш  $i:=100$ ,  $S1:=1$ ; токи  $i \geq 1$  такрорлаш бошланиш  $S1:=S1+i$ ;  $i:=i-1$  тамом; чикариш  $S1$ ; тамом.

2-алгоритм: бошланиш  $i:=100$ ,  $S2:=1$ ; токи  $i \geq 1$  такрорлаш бошланиш  $S2:=S2*i$ ;  $i:=i-1$  тамом; чикариш  $S2$ ; тамом.

Биринчи ва иккинчи алгоритм бажарилиши натижасида мос равишда  $S1$  ва  $S2$  кийматлар ҳосил қилинади.  $S1$  ва  $S2$  уртасида куйидаги келтирилган муносабатлардан қайси бири бажарилади?

A)  $S1 < S2$ ;

B)  $S1 > S2$ ;

C)  $S1 = S2$ ;

D)  $S1 = 2 * S2$ ;

28. Куйидаги  $(3+4*8 > (15.5-2) \bmod 3 = \text{true})$  ифодани ҳисоблашда амалларнинг бажарилиш тартибини аниқланг:

A) келтирилган жавоблар орасида тугриси йук.

B) \*, mod, +, -, >, =

C) \*, >, +, mod, -, =

D) \*, +, >, mod, -, =

29. Паскаль тилидаги дастур булимларининг жойланиш тартиби қандай булиши керак?

A)

1. Нишонлар булими;
2. Узгармаслар булими;
3. Турлар булими;
4. Узгарувчилар булими;
5. Процедура ва функциялар булими;
6. Операторлар булими;

B)

1. Турлар булими;
2. Узгарувчилар булими;
3. Операторлар булими;
4. Нишонлар булими;

C)

1. Нишонлар булими;
2. Узгармаслар булими;
3. Турлар булими;
4. Операторлар булими;
5. Узгарувчилар булими;
6. Процедуралар ва функциялар булими;

D)

5. Узгарувчилар булими;
6. Процедуралар ва функциялар булими;
1. Нишонлар булими;
2. Турлар булими;
3. Узгармаслар булими;
4. Операторлар булими;
5. Процедура ва функциялар булими;
6. Узгарувчилар булими

30. Берилган  $\frac{x^2 + 3 - y}{A * \sin X + e^y}$  ифодани Паскал тилида ёзиш талаб қилинади. Қуйидаги ёзувлардан қайси бири тугри?

- A)  $(X * X + 3 - Y) / (A * \sin(X) + \exp(Y))$ ;
- B)  $\text{SQR}(X) + 3 - Y / A * \sin(X) + \exp(Y)$ ;
- C)  $X * X + (3 - Y) / (A * \sin(X) + \exp(Y))$ ;
- D)  $(X * X + 3 - Y) / (A * \sin(X) + \exp(Y))$ ;

31. Қуйидагиларнинг қийматини топинг:

- 1) ORD (CHR(49));                      2) CHR(ORD('\*\*'));
- A) 1) 49; 2) '\*\*'
  - B) 1) '\*\*'; 2) 49
  - C) 1) 49; 2) 'A'
  - D) 1) 50; 2) '\*\*'

32. INTEGER турига тегишли узгарувчилар учун амалларнинг қуйида келтирилган гуруҳларидан (яъни А, В, С, D, Е - гуруҳлардан) қайси бири тулик аниқланган:

- A) +, -, \*, /, DIV, MOD, =, <, <=, >, >=;
- B) -, MOD, DIV, OR, +, \*;
- C) =, <, <=, >, >=, AND, OR, NOT;
- D) AND, OR, NOT, +, -, \*, /, MOD, DIV

33. Белги (CHAR) туридаги узгарувчилар устида қандай амалларни бажариш мумкин?

- A) муносабат амаллари: =, <, <=, >, >=;
- B) мантикий амаллар : AND, OR, NOT,
- C) арифметик ва мантикий амаллар;
- D) муносабат амаллари ва арифметик амаллар.

34. Мантикий турдаги узгарувчилар устида қайси амалларни бажариш мумкин.

- A) мантикий амаллар ҳамда муносабат амаллари;
- B) фақат муносабат амаллари: =, <>, <, >, <=, >=;
- C) фақат мантикий амаллар: and, or, not,
- D) арифметик амаллар ҳамда мантикий амаллар;

35. Куйидагиларнинг кийматини топинг:

- 1) pred ('b');            2) succ ('c');
- 3) 15 mod 4;            4) 22 div 4;

- A) 1) 'a'; 2) 'd'; 3) 3; 4) 5;
- B) 1) 'd'; 2) 'e'; 3) 5; 4) 4;
- C) 1) 'c'; 2) 'b'; 3) 2; 4) 5;
- D) 1) 'c' 2) 'b'; 3) 5; 4) 2;

36. Куйида келтирилган ифодаларан қайси бири Паскаль тили қодаларига зид келмайди?

- A) '0' or '9' ;
- B) (15>true)\*q;
- C) SIN (3.14)+COS (32.1)\* SQR (7.6)-0.1E-5;
- D) LN (55)\*false> ORD (':');

37. Куйида узгарувчиларни тасвирлашга доир мисоллар келтирилган. Улардан қайси бири хатосиз езилган?

- A) Var sum: real; last: integer;  
**Lambda: char; kara1: boolean**
- B) Var s1, nnNN, x [50]: char;  
f,y,a: integer;
- C) Var true, false: boolean;  
a, A1, max:real;
- D) Var coef, k15: boolean;  
var: char;

38. Куйидаги жадвалда бутун турга тегишли аргумент билан ишлатилиши мумкин булган стандарт функцияларнинг номи, математик киймати ва натижа тури берилган

№	Функция номи	Математик киймати	Натижа тури
1	ARCTAN (X)	arctan x	real
2	SUCC (X)	x+1	integer
3	PRED (X)	x-1	real

Бу жадвалда фақат битта хато мавжуд. Уни тугрилаш учун куйидаги ёзувлардан қай бирини ишлатиш керак?

- A) integer

- B) true
- C) sqrt (x)
- D) real

39. Берилган: S,C1,C2 - char турига, I эса integer турига тегишли узгарувчилар. Куйидаги бандлардан кайси бирида хатога йул куйилган?

A) **ORD (CHR(I))=I**

- B) CHR (ORD) (S)=S
- C) C1>C2 муносабат факат ORD (C1)<ORD (C2) булгандагина уринли булади.
- PRED (S)= CHR (ORD(S)-1)

40. Куйидаги бандлардан кайси бирида хатога йул куйилган?

- A) Мантикий константалар учун FALSE<TRUE тенгсизлик уринли.
- B) мантикий турдаги узгарувчилар устида куйидаги амаллар аниқланган: AND, OR, NOT.
- C) мантикий турга тегишли узгарувчилар TRUE (чин) ва FALSE (ёлгон) кийматлардан бирини кабул килиши мумкин.
- D) PRED (TRUE)= FALSE;

41. Куйидаги санок скаляр турларни тавсифлаш ва уларга тегишли узгарувчилар устида амаллар бажаришга доир мисоллар келтирилган. Бу мисоллардан кайси бири хатосиз ёзилган?

A) Type T1=(AMAD, CAMAD, BYRI, ALI);  
T2=(OQ, QORA, KUK, KIZIL);  
VAR X,Y:T1;A,B:T2;  
X:ALI; A:=KUK; B:=OQ

B) Type T1=(MEN, CEN, Y, 0.5);  
T2=(INB, FEV, MART, APR, MAI, JUH);

C) Type T1 =(KATTA, KICHIK, URТА);  
T2=(STOL,STUL,PARTА);  
VAR X,Y,:T1;A,B:T2;  
X:STOL;Y:=KICHIK;T2:=URТА;

D) Type T1 =(KATTA, KICHIK, URТА);  
T2=(STOL,STUL,PARTА);  
VAR X,Y: boolean;  
X:=KATTA<KICHIK;  
Y:=STUL>URТА;

42. A,B,C,D узгарувчилар куйидагича тасвирланган:

Var A: integer;  
B: real;  
C: boolean;  
D: char;

Куйидаги таъминлаш операторидан кайси бири тугри?

- A) B:=A+sqrt(5);
- B) A:=true;
- C) C:=false+3>4;
- D) D:=A/B+A\*B;

43. Куйидаги таъминлаш операторларидан кайси бири тугри?

- A) Z895A:=abs(x)+sqr(3.14)\*0.5;
- B) 1YN:=3<8;
- C) S+5:=sqr(9.12\*(45+sqr(965)));
- D) Хаммаси тугри

44. Таъминлаш оператори кандай ишни бажариш учун мулжалланган? Энг умумий жавобни топинг.

- A) Операторнинг унг кисмида турган ифодани хисоблайди ва унинг кийматини чап кисмдаги узгарувчига таъминлайди.
- B) Узгарувчиларга киймат таъминлайди.
- C) Узгарувчиларнинг турини бошкасига узгартиради.
- D) Ифода киймати кайси турга мансублигини аниқлайди.

45. Куйидаги киритиш операторлари учун маълумотлар экранга кандай берилади:  
readln (x,y,z); read (a,b,c);

- A) x, y ва z бир каторда, a,b,c лар эса навбатдаги каторда берилган булиши мумкин
- B) x, y лар бир каторда; z, a лар – иккинчи каторда; b, c лар – учинчи каторда берилиши керак
- C) Хамма узгарувчилар киймати бир каторда жойлашмоги зарур.
- D) x, y, z лар бир ёаторда, a – иккинчи ёаторда, b – учинчи ёаторда, c – туртинчи ёаторда

46. Куйидаги ёзувлардан кайси бири тугри?

- A) Read (a,d); Readln (c,x);
- B) Readln (a\*d,c,x);
- C) Readln (Sqr(x));
- D) Readln; Read (a); Readln (b,c+3,x);

47. Чикариш операторини оддий холда шундай ёзиш мумкин:

Write(V1, V2,...,VN);

Бу ерда V1, V2,...,VN лар урнида нималар келиши мумкин?

- A) узгармаслар;
- B) Узгармаслар хамда нишонлар;
- C) Файл номи;
- D) Ихтиёрий турдаги ифодалар;

48. X, YX, Z1 ва A узгарувчиларнинг киймати берилган: X=125; YX=3,14; Z1='\$'; A=true. Экраннынг бир сатрида “3.14 true 125\$” ёзувни хосил килиш талаб килинади. Куйида келтирилган операторлардан кайси бири шу ишни бажара олади?

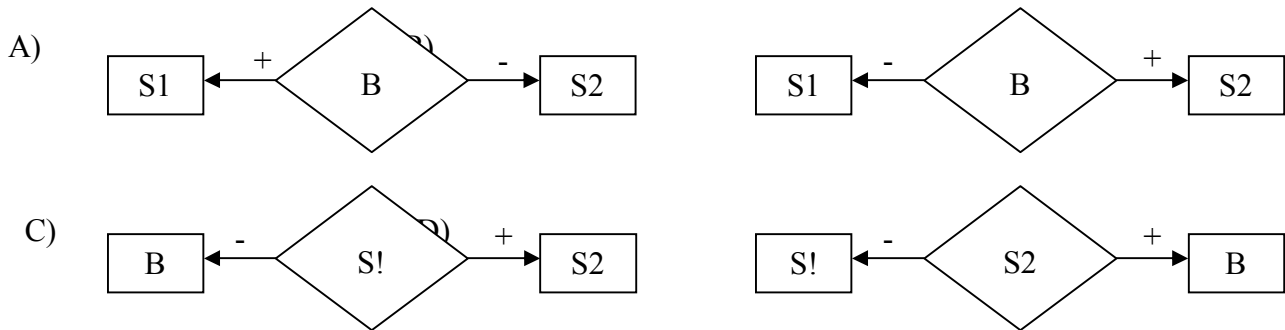
- A) Writeln; Writeln (YX:4:2, ' ', A, ' ', X:3, Z1);
- B) Write (X,Yx,Z1,A:4);
- C) Write (YX:6:2,A);
- D) Write (' ', Z1, ' ', A, ' ', X:3);

49. Шартли операторнинг умумий куруниши куйидагича If S1 then S2 else S3 бу ерда S1, S2 ва S3 лар нимани билдиради?

- A) S1 - мантикий ифода; S2 ва S3 - Паскаль тилининг ихтиёрий операторлари;
- B) S1 - ихтиёрий ифода; S2 ва S3 - Паскаль тилининг ихтиёрий операторлари;

- C) S1 - арифметик ифода; S2 ва S3 - Паскаль тилининг ихтиёрый операторлари;  
 D) S1 - логик ифода; S2 – шарт; S3-ихтиёрый оператор;

50. Шартли оператор If B then S1 else S2 нинг бажарилишига мос блок-схемани курсатинг.



## 11. BAHOLASH MEZONLARI VA REYTING NAZORATLARI GRAFIGI

### 11.1. BAHOLASH MEZONLARI

«Algoritmlar nazariyasi» fani bo'yicha joriy nazoratlarda talabalar bilimi va amaliy ko'nikma darjasini aniqlash mezonlari  
(max ball – 35)

Maksimal ball		Nazorat qilinadigan va baxolanadigan ish turlari	Baxolashda e'tibor karatiladigan jixatlar
1-jn	2-jn		
3	4	Mavzular bo'yicha nazariy tayyorgarlik darajasi va darsdagi faollik	Asosiy tushunchalar, ta'riflar, algoritmlarni tuzish, algoritmlarni amalga oshirish usullarini bilish, mohiyatini tushunish, ijodiy fikrlay olish, bilimlarni amalda qullay olish
3	4	Uyga berilgan topshiriqlarni bajarish sifati	Topshiriqlarni to'g'ri va to'liq bajarish, masalalarni hal qilishga ijodiy yondashish, tushuntirib bera olish
6	8	Nazorat ishlarini bajarish sifati	Topshiriqlarni to'g'ri va to'liq bajarish, ijodiy yondashish, mustaqil fikrlash, yechimni asoslay olish
3	4	Mustaqil topshiriqlarni bajarish sifati	Berilgan topshiriqni to'g'ri va to'liq bajarish, mustaqil muloxaza yurita olish, bilimlarni amalda qo'llay olish, masalaga ijodiy yondashish, mohiyatini tushunish va aytib bera olish
15	20		

«Algoritmlar nazariyasi» fani bo'yicha oraliq va yakuniy nazoratlarda talabalar bilimi va amaliy ko'nikma darjasini aniqlash mezonlari

**(ON bo'yicha max ball – 35, YaN bo'yicha max ball – 30)**

Savollar		ON (max ball)		YaN (max ball)	Baxolashda e'tibor qaratiladigan jixatlar
		1-ON	2-ON		
Nazariy	1	3	4	6	Asosiy tushunchalar, ta'riflar, algoritmlarni tuzish, algoritmlarni amalga oshirish usullarini bilish, tasavvur qilish va aytib bera olish, ijodiy fikrlay olish va mustaqil muloxaza yurita olish
	2	3	4	6	
Amaliy	3	3	4	6	Topshiriqlarni to'g'ri va to'liq bajarish, ijodiy yondashish, mustaqil fikrlash, yechimni asoslay olish, mohiyatini tushunish
	4	3	4	6	
Mustaqil ish	5	3	4	6	Savolga to'liq va to'g'ri javob berish, misollar bilan asoslash, ijodiy yondashish, mohiyatini tushunish va tushuntirib bera olish
Jami		15	20	30	



«Algoritmlar nazariyasi» fani bo'yicha reyting nazoratlarida o'zlashtirish ko'rsatkichini aniqlash mezonlari

JN	ON	YaN	Baxolashlarda e'tibor qaratiladigan asosiy jihatlar
31-35 ball	31-35 ball	27-30 ball	Asosiy tushuncha, ta'rif, algoritmlarni tuzish, algoritmlarni amalga oshirish usullarini bilish, mohiyatini tushunish, ijodiy fikrlay olish, tasavvurga ega bo'lish, aytib bera olish, mustaqil muloxaza yurita olish, topshiriqlarni aniq va to'g'ri bajarish
25-30 ball	25-30 ball	22-26 ball	Asosiy tushuncha, ta'rif, algoritmlarni tuzish usullarini bilish, ijodiy yondashishga xarakat qilish, tasavvurga ega bo'lish, topshiriqlarni to'g'ri bajarish va tushuntirish
19-24 ball	19-24 ball	17-21 ball	Asosiy tushuncha, ta'rif, algoritmlarni bilish va amalda qo'llay olish, mohiyatini biroz tushunish va to'liq bo'lmagan tasavvurga ega bo'lish. Amaliy topshiriqlarni deyarli to'g'ri bajarish va tushuntirib berishga xarakat qilish.
0-19 ball	0-19 ball	0-15 ball	Asosiy tushuncha, ta'riflarni to'liq bilmaslik va amalda qo'llay olmaslik, yetarlicha tasavvurga ega bo'lmaslik va tushuntira olmaslik, topshiriqlarni to'liq bajarmaslik va qo'pol xatoliklarga yo'l quyish.

## 11.2. REYTING NAZORATLARI GRAFIGI

*SAMARQAND DAVLAT UNIVERSITETI*  
*MEXANIKA-MATEMATIKA FAKULTETI*  
 «Axborotlashtirish texnologiyalari» kafedrası  
 «Algoritmlar nazariyasi» fani bo'yicha reyting nazoratlari  
 GRAFIGI

*Mta'lim yo'nalishi: Amaliy matematika va informatika (2-kurs)*

Umumiy o'quv soati-126 soat, shundan ma'ruza-30 soat, amaliyot – 32 soat, mustaqil ish - 54 soat.

2010-2011 o'quv yili  
1 semestr

Ishchi o'quv dasturidagi mavzular tartib raqami	Umumiy soat					Baxolash turi	Nazorat shakli	Ball		Muddati (xafta)
	Ma'ruza	Amal.mashg'ulot	Labarotoriya	Mustaqil ish	Jami			Maks. ball	Saral. ball	
1-7	14	14		24	52	1-JB	Dars jarayonida nazorat, uy ishi va davomat	15		dekabr 2-xafta
						1-OB	Yozma nazorat	15		dekabr 2-xafta
8-15  Qo'shimcha mavzu bo'yicha referat	16	18		30	64	2-JB	Dars jarayonida nazorat, uy ishi va davomat	20		yanvar 3-xafta
						MB	Ximoya			
						2-OB	Yozma nazorat	20		yanvar 3-xafta
Jami	30	32		54	126	JB		35		
						OB		35		
						Jami		70	39	
						YaB	yozma	30		Fevral jadval bo'yicha
Jami						100	55			

## 11. MA'RUZA MASHG'ULOTLARI DARS ISHLANMASI

Alisher Navoiy nomidagi

Samarqand Davlat universiteti

“Axborotlashtirish texnologiyalari” kafedrası

Axatov A.R.

**Algoritmlar nazariyasi fanidan**

*ma’ruza masg’ulotlar ishlanmasi*

**SAMARQAND – 2010**

# 1 - MAVZU: KIRISH. ALGORITMLASH FANI VA ALGORITMLASH SAN'ATI

## *Reja*

- 1. Algoritm tushunchasi.**
- 2. Algoritmni tuzish – ijodiy ish**
- 3. Algoritm sifati baholash.**

**Darsning maqsadi:** talabalarga algoritm tushunchasi, zamonaviy matematika va informatikadagi o'rni, algoritmni tuzish – ijodiy ish ekanligini ko'rsatish.

**Tayanch iboralar:** algoritm nazariyasi, algoritmni tuzish, masala quyilishi, modellashtirish, test, ishlab chiqish, hujaatlashtirish.

**Mashg'ulot vositalari:** sinf doskasi, plakatlar, fundamental fan darsliklari, o'quv va uslubiy qo'llanmalar, informatika bo'yicha atamalar lug'ati, videoprojektor, ekran va kompyuterdan samarali foydalanish.

**Mashg'ulot usullari:** takrorlash, suhbat va savol-javob, munozara (mavzuni o'zlashtirishni mustahkamlash) tarzida muloqot o'tkazish, (talabalarning mustaqil, erkin fikrlash va so'zlashga o'rgatgan holda fikr mulohazalarini bayon qildirish, buning uchun har bir talabaga, tayanch iboralardan savollar tashlanadi, ular o'z fikrlarini bayon qiladilar, hamma talaba javobni bayon qilib bo'lgandan so'ng talaba bilan birgalikda javoblar yakun qilinadi).

**Darsning xronologik xaritasi** – 80 minut.

**Tashkiliy qismi:** Auditoriyaning jixozlanishi va sanitar sharoitlari, talabalar davomati – 2 minut.

**Bilimlarni baholash:** yangi mavzuni o'rganish uchun zarur bo'lgan material bo'yicha suxbat – 10 minut.

**Yangi mavzuni bayon etish** – 55 minut.

**Mavzu o'zlashtirilgan darajasini aniqlash** – 10 minut.

**Uyga vazifa** – 3 minut.

Algoritm tushunchasi zamonaviy matematika va informatikaning asosiy tushunchalaridan biri hisoblanadi. Algoritm termini o'rta asrlar ulug' matematigi al-Xorazmiy nomidan kelib chiqqan. XX asrning 30-yiligacha algoritm tushunchasi ko'proq matematik ma'no emas, balki metodologik ma'noni kasb etar edi. Algoritm deganda, u yoki bu masalalar sinfini yechish imkonini beruvchi aniq ifodalangan chekli qoidalar majmui tushunilgan. EHM larning paydo bo'lishi bilan algoritm tushunchasi yanada keng tarqaldi. EHM va dasturlash usullarining rivojlanishi algoritmni ishlab chiqish avtomatlashtirishdagi zaruriy bosqich ekanligini tushunishga yordam berdi. EHM larning paydo bo'lishi algoritm nazariyasining rivojlanishiga olib keldi.

Algoritmni tuzish – bu ijodiy ish bo'lib, ixtiyoriy zaruriy algoritmni tuzish uchun umumiy usullar mavjud emas, kishining ijodiy qobiliyatiga bog'liq.

Albatta, algoritmni aniq sxema bo'yicha tuzish zarur bo'lib qoladigan sodda hollar ham mavjud. Bunday hollarda yechilish algoritmi avval biron kim tomonidan olingan masalalarni misol keltirish mumkin. Masalan, differensial tenglamalarni sonli integrallash uchun Eyler metodi. Bu metod masalani yechish uchun umumiy holda ifodalangan algoritmdir, lekin algoritmni ijodiy ekanligini quyidagi algoritm nazariyasining ba'zi bir ma'lumotlaridan ko'rish mumkin.

Agar bizdan biror algoritmni ishlab chiqish talab qilinsa, dastlab izlanayotgan algoritmni tuzish mumkinmi yo'qmi degan savolga javob izlash kerak. Chunki ba'zi hollarda algoritmni tuzish mumkin emasligini ko'rsatib berish mumkin. Ba'zi bir hollarda algoritmni tuzish mumkinligi isbotlanadi. Bunday isbot mavjud bo'lganligi bilan tuzilgan algoritmni amalgam oshirib bo'lmaydi yoki uning samaradorligi talabga javob bermaydi. Shunga qaramasdan bir nechta algoritm bitta amaliyotga qo'llanilayotganini topish mumkin.

Boshqa hollarda algoritmni tuzish mumkinligini ham, mumkin emasligini ham isbotlab bo'lmaydi. U vaqtda algoritm tuzish jarayonida boshqa predmet sohalaridan qurilgan algoritmlardan foydalanish mumkin.

Algoritm sifati baholash uchun mezonlarni ko'raylik. Mavjud mezonlar juda tahminlashgan. Masalan, algoritmni bajarishda bajaruvchining xotira uskunalari hajmi yetarli bo'lmasa, u algoritm yomon deb hisoblanadi. Boshqa mezon sifatida algoritmning bajarilishi uchun talab qilinadigan vaqtni ko'rsatish mumkin. Vaqtni baholash bajaruvchining fizik xarakteristikalari hisobga olinishi kerak. Chunki har bir operatsiya har xil o'zgaruvchilar bilan bajarilganda vaqt ham har xil bo'ladi. Bunchalik aniq ma'lumotni har bir foydalanuvchi uchun yig'ib bo'lmaganligi sababli odatda o'rtacha tezkorlik qabul qilinadi. Ketma-ket bajarilayotgan operatsiyalar sonini aniqlab, uni o'rtacha tezkorlikka ko'paytirsa, algoritm bajarilishining amalga yaqin bo'lgan vaqtini topishimiz mumkin.

Faraz qilaylik, 2 ta tahlil qilingan algoritmlardan bittasining bajarilish vaqti tezroq bo'ladi, uni xotira ishlash hajmi bo'yicha ham tahlil qilish kerak va bunday tahlillar murakkab nazariyasiga mansub bo'ladi. Shunday qilib, algoritm nazariyasi fani masalalarni yechishga mo'ljallangan algoritmlarni samaradorligini va murakkabligini tahlil qilish, o'zgartirish, qo'shimcha qilish va qayta ishlash natijasida yaxshilash usul va uslublarini o'rganadi.

### **Takrorlash ucun savollar**

1. Algoritm tushunchasi.
2. Algoritm nazariyasining paydo bo'lishi
3. Nimaga algoritmni tuzish ijodiy ish hisoblanadi?
4. Algoritmni va ularning murakkabligini tahlil qilishda nimalarga e'tibor berish kerak?

### **Mustaqil ishlash uchun nazorat savollari:**

1. Algoritm tushunchasiga misol ko'rsating.
2. Algoritm so'zi kelib chiqishi haqida ma'lumot bering
3. Zamonaviy algoritmlarga nisbatan mavjudlardan tashqari qanday hossalarni qo'shish mumkin?
4. Algoritmni kunlik hayotimizda qo'llashga misol ko'rsating.
5. Dasturni tekshirish uchun kiritiladigan test ma'lumotlariga misol tuzing.

### **Mavzuga doir testlar:**

1. Куйидаги бандлардан кайси бирида алгоритм тушунчаси аниқроқ ва тулиқроқ таърифланган?

А) Алгоритм-куйилган масалани ечиш ёки маълум бир мақсадга эришиш учун ижрочи бажариши зарур булган иш харакатнинг (амалларнинг) тушунарли ва аниқ кетма-кетлигидир.

В) Алгоритм узбек математиги Ал Хоразмий номи билан боғлиқ булиб, унинг европача бузиб айтилишидир.

С) Алгоритм деганда ЭХМ учун тузилган дастурни тушунамиз.

Д) Алгоритм ижроцига берилган курсатма (йурикнома) булиб хизмат килади.

2. Quyidagi hossalardan qaysi biri bo'yicha algoritmning har bir qadami aniq belgilangan bo'lishi kerak?

A) Aniqlik

C) Moslashuvchanlik

B) Ommaviylik

D) Diskretlik

3. Algoritmning birinchi rasmiy tushunchasini kim kiritgan?
- A) Gedel  
B) Muxammad Al-Xorazmiy  
C) Tyuring  
D) Paskal
4. Algoritm tushunchasini intuitiv darajada birinchi bo'lib kim kiritgan?
- A) Muxammad Al-Xorazmiy  
B) Paskal  
C) Tyuring  
D) Gedel
5. Bugungi kunda eng keng tarqalgan algoritmning rasmiy ta'rifi muallifi kim?
- A) Tyuring  
B) Muxammad Al-Xorazmiy  
C) Paskal  
D) Fon-Neymanl

### Adabiyotlar

1. В.А.Успенский, А.Л.Семенов. Теория алгоритмов: основные открытия и приложения. – М: Наука, 1987, 287 с.
2. Т.Кормен, Ч.Лейзерсон, Р.Ривест. Алгоритмы: построение и анализ. Сер: Классические учебники. М.: МЦНМО, 2001.- 960 с.
3. Гуломов С.С. ва бошқалар. Ахборот тизимлари ва технологиялари. Тошкент, 2000 й.
4. Жуманов И.И. Мингбаев Н.С., Информатика.- Самарқанд,: СамДУ нашри, 2002, 107 бет.
5. Ahatov A.R., Zaripova G.L. va boshq. Axborot texnologiyalari // Uslubiy qo'llanma. – Samarqand: SamDU nashri, 2008 yil – 112 bet.

## 2 - MAVZU: ALGORITMLASHNING MATEMATIK ASOSLARI

### Reja

1. *Matematik induksiya.*
2. *Yig'indi va Ko'paytmalar.*
3. *Butun qiymatli funksiyalar.*
4. *O'rin almashtirishlar va faktoriallar.*
5. *Binomial koeffitsiyentlar.*
6. *Fibonachi sonlari.*

**Darsning maqsadi:** talabalarga algoritmlashning matematik asoslarini, zamonaviy matematika va algoritmlash bog'lanishi, algoritmlashda qo'llaniladigan asosiy matematik funksiyalar haqida ma'lumot berish.

**Tayanch iboralar:** algoritmlar nazariyasi, algoritmni tuzish, masala quyilishi, modellashtirish, test, ishlab chiqish, hujaatlashtirish.

**Mashg'ulot vositalari:** sinf doskasi, plakatlar, fundamental fan darsliklari, o'quv va uslubiy qo'llanmalar, informatika bo'yicha atamalar lug'ati, videoprojektor, ekran va kompyuterdan samarali foydalanish.

**Mashg'ulot usullari:** takrorlash, suhbat va savol-javob, munozara (mavzuni o'zlashtirishni mustahkamlash) tarzida muloqot o'tkazish, (talabalarning mustaqil, erkin fikrlash va so'zlashga o'rgatgan holda fikr mulohazalarini bayon qildirish, buning uchun har bir talabaga, tayanch iboralardan savollar tashlanadi, ular o'z fikrlarini bayon qiladilar, hamma talaba javobni bayon qilib bo'lgandan so'ng talaba bilan birgalikda javoblar yakun qilinadi).

**Darsning xronologik xaritasi** – 80 minut.

**Tashkiliy qismi:** Auditoriyaning jixozlanishi va sanitar sharoitlari, talabalar davomati – 2 minut.

**Bilimlarni baholash:** yangi mavzuni o'rganish uchun zarur bo'lgan material bo'yicha suxbat – 10 minut.

**Yangi mavzuni bayon etish** – 55 minut.

**Mavzu o'zlashtirilgan darajasini aniqlash** – 10 minut.

**Uyga vazifa** – 3 minut.

Algoritmarni tuzishda va ularning tahlilida ishlatiladigan ba'zi matematik belgilashlarni qarab chiqamiz.

### Matematik induksiya.

Faraz qilaylik  $P(n)$  – bu  $n$  butun son to'g'risidagi biror bir tasdiq bo'lsin. « $n(n+3)$  – juft son»  $n \geq 10$  bo'lsa, u holda  $2^n > 2^{3n}$ . Bizdan  $P(n)$  ning barcha butun musbat  $n$  sonlar uchun o'rinli ekanligini isbotlash talab qilinsin. Isbotning asosiy usuli quyidagilardan iborat:

3.  $P(1)$  o'rinli ekanligini isbotlash.

4.  $P(1), P(2), \dots, P(n)$  lar o'rinli bo'lsa, u holda  $P(n+1)$  ham o'rinli ekanligini isbotlash, bu isbot barcha butun musbat  $n$  lar uchun o'rinli bo'lishi kerak.

**Misolni keltiramiz.**

$$1 = 1^2 \quad 1 + 3 = 2^2 \quad 1 + 3 + 5 = 3^2 \quad 1 + 3 + 5 + 7 = 4^2 \quad (1)$$

Ularning umumiy ko'rinishda quyidagicha yozish mumkin:

$$P(n) = 1 + 3 + \dots + (2n - 1) = n^2 \quad (2).$$

Biz  $P(n)$  ning barcha musbat  $n$  lar uchun o'rinli ekanini isbotlashimiz kerak. Yuqoridagi proseduraga muvofiq:

a).  $P(1)$  o'rinli, chunki  $1 = 1^2$

b). agar barcha  $P(1), P(2), \dots, P(n)$  tasdiqlar o'rinli bo'lsa,  $P(n)$  uchun ham o'rinli, ya'ni (2) munosabat bajariladi.

(2) ning har ikkala tomoniga  $2n+1$  ni qo'shsak, quyidagiga ega bo'lamiz:

$$1 + 3 + 5 + \dots + (2n - 1) + 2n + 1 = n^2 + 2n + 1 = (n + 1)^2$$

Bu esa  $P(n+1)$  ning ham to'g'riligini ko'rsatadi.

Bu metodni isbotlashning algoritmik prosedurasi deb qarash mumkin. Haqiqatan ham, agar a) va b) bosqichlar amalga oshgan deb hisoblasak, quyidagi algoritm  $P(n)$  tasdiqning ixtiyoriy butun musbat  $n$  uchun isbotini beradi.

Berilgan butun musbat  $n$  uchun  $P(n)$  ning o'rinli ekanini isbotlash algoritmi.

A1 algoritmi.

7. boshlash.

8.  $k \rightarrow 1$  {((a)ga asosan  $P(1)$  tasdiqni isbotlang}

9. agar  $k=n$  bo'lsa, u holda 6 ga o'ting

10.  $p(k+1)$  uchun isbotlang ((b) ga asosan  $p(2), p(3), p(k)$  to'g'riligini isbotlang va  $p(k+1)$  uchun to'g'ri degan xulosaga keling)

11.  $k \leftarrow k + 1$  3 ga o'ting

12. tugash (so'ralayotgan isbot bajarildi)

(b) va (b) bosqichlar (a1 algoritmi) shaklidagi isbotlash matematik induksiya yordamida isbotlashdir

### Yig'indi va Ko'paytmalar.

$a_1, a_2, \dots$  - ixtiyoriy sonlar ketma-ketligi bo'lsin.  $a_1 + a_2 + \dots + a_n$  ko'rinishdagi yig'indini  $\sum_{i \leq j \leq n} a_j$  kompakt ko'rinishida yozish mumkin.

Agar  $n$  nolga yoki manfiy songa teng bo'lsa berilishiga ko'ra bu yig'indi nolga teng bo'ladi.  $j$  harfi indeks yoki yig'indining o'zgaruvchisi.

Yig'indilar chekli (j qiymatlarini chekli soni) va cheksiz bo'lishi ham mumkin. Agar  $\sum$  belgisi ostida ikki yoki undan ortiq shartlar joylashgan bo'lsa, ularning barchasi bir vaqtning o'zida bajarilish kerak.

Yig'indi uchun qisqa yozuv bo'lganidek, ko'paytma uchun ham  $\prod_{1 \leq j \leq n} a_j$  qisqa yozuv ishlatiladi.  $\prod_{1 \leq j \leq n} a_j$  belgi  $1 \leq j \leq n$  shartni qanoatlantiruvchi barcha butun j lar uchun barcha  $a_j$  lar ko'paytma 1ga teng deb hisoblanadi (yig'indi esa nolga teng bo'ladi).

### **Butun qiymatli funksiyalar.**

Ixtiyoriy haqiqiy son uchu quyidagi belgilashlarni kiritamiz:

$\lfloor x \rfloor$  - x ga eng yoki x dan kichik bo'lgan eng katta butun son.

$\lceil x \rceil$  - x ga eng yoki x dan katta bo'lgan eng kichik butun son.

Bu funksiyalar ni ba'zida x sonining butun qismi deb yuritiladi.

Masalan:  $\lfloor \sqrt{2} \rfloor = 1$   $\lceil \sqrt{2} \rceil = 2$   $\lfloor x \rfloor = \lceil x \rceil$ .

Ixtiyoriy haqiqiy x va y sonlar uchun quyidagi Binar amalini belgilaymiz.  $X \bmod Y$  - x ni y ga bo'lgandagi qoldiqni bildiradi. Agar x va y lar butun son bo'lsa, u holda qoldiq ham butun son va x,y ga karrali bo'lsa, nol bo'ladi.

$$5 \bmod 3 = 2$$

$$18 \bmod 3 = 0$$

Agar x va y butun sonlar bo'lsa, div butun qiymatli bo'lishni bildiradi, ya'ni butun qiymatli bo'lish natijasida har doim butun bo'ladi.

$$7 \operatorname{div} 2 = 3$$

$$2 \operatorname{div} 5 = 0$$

### **O'rin almashtirishlar va faktoriallar.**

n tartibli o'rin almashtirish deb, n ta turli ob'yektlarni qatorga joylashtirish operatsiyasiga aytiladi.

Masalan, a, b, c lar uchun 6 ta o'rin almashtirishlar bor. abc, bac, bca, cba, cab, acb.

n ob'yektdan tuzish mumkin bo'lgan umumiy o'rin almashtirishlar soni

$$P(n) = n(n-1)(n-2) \dots 1 = n!$$

P(n) qiymatni n! deb hisoblaydilar va u quyidagicha yoziladi.

$$n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n = \prod_{1 \leq k \leq n} k$$

$0! = 1$  ekanligi qabul qilingan. Butun musbat n lar uchun  $n! = (n-1)!n$  ayniyat o'rinli.  $0! = 1$   $1! = 1$   $3! = 6$ .

Faktoriallar juda tez o'sadi.  $10! = 3628800$

$1000!$  esa 2500 dan ortiq o'nli belgilardan iborat. Shunga qaramasdan kompyuterda faktorialni hisoblash uchun kam vaqt ketadi.

Dj. Stirling degan olim  $n! \approx \sqrt{2\pi n} \left(\frac{n}{e}\right)^n$  ga teng deb olgan.

Yana bir savol tug'ildi. Biz n! uchun n butun musbat bo'lgan hol uchun ta'rif berdik. n ning ratsional qiymatlari yoki n haqiqiy bo'lganda n! nimaga teng degan savol tug'iladi. Masalan,

$\left(\frac{1}{2}\right)!$  nimaga teng. Bu masalani yechish uchun butun manfiymas n lar uchun n! ni aniqlaymiz.

$$n! = 1 + 2 + \dots + n = \sum_{1 \leq k \leq n} k \quad (1)$$

Bu faktorialning analogi, lekin bu yerda biz ko'paytirish o'rniga qo'shishdan foydalanayapmiz

$$\text{Arifmetik progressiyaning yig'indisi} \quad n! = \frac{1}{2} n(n+1) \quad (2)$$



(2) ni (1) ning o'rniga ishlatish  $n!$  funksiyani  $n$  ning ixtiyoriy qiymatlari uchun aniqlash imkonini beradi. Masalan,  $\left(\frac{1}{2}\right)! = \frac{3}{8}$ .

### **Binomial koeffitsiyentlar.**

$n$  ta ob'yektdan  $k$  ta ob'yektni jamlash bu  $n$  ta elementdan mumkin bo'lgan  $k$  ta turli elementni tanlash. Masalan, 5 ob'yektdan 3 tadan jamlash, a, b, c, d, e. abc, abd, abe, acd, ace, ade, bcd, bce, bde, cde.

$n/k$  orqali belgilangan jamlashni umumiy soni

$$n/k = \frac{n(n-1)\dots(n-k+1)}{k(k-1)\dots 1}$$

Masalan  $\left(\frac{5}{3}\right) = \frac{5 \cdot 4 \cdot 3}{3 \cdot 2 \cdot 1} = 10$ .

$\left(\frac{n}{k}\right)$  qiymat binomial koeffitsiyent deb aytiladi. Binomial koeffitsiyentni faktorial yordamida

hisoblash mumkin.  $\left(\frac{n}{k}\right) = \frac{n!}{k!(n-k)!}$

Binomial koeffitsiyentlar uchun quyidagi hossa mavjud:

$$\left(\frac{r}{k}\right) = \frac{r}{k} \left(\frac{r-1}{k-1}\right)$$

### **Fibonachi sonlari.**

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55,...

ketma-ketlikda har bir son oldingi 2 ta sonning yig'indisiga teng bo'lsa, Fibonachi sonlari deb aytiladi.

$$F_n = F_{n-1} + F_{n-2} \quad n > 0.$$

Bu ketma-ketlik Leonardo Fibonachi tomonidan taklif etilgan. Fibonachi sonlari va algoritmlar orasida o'zaro bog'liq borligi isbotlangan.

Takrorlash uchun savollar

1. Matematik induksiya haqida tushuncha bering.
2. Yig'indi va Ko'paytmalarning asosoy farqini ko'rsating.
3. Butun qiymatli funksiyalarga misol keltiring.
4. O'rin almashtirishlar va faktoriallarni hisoblashga misol ko'rsating.
5. Binomial koeffitsiyentlar - bu nima?
6. Fibonachi sonlari algoritmlarga qanday aloqasi bor?

### **Mustaqil ishlash uchun nazorat savollari:**

1. Matematik induksiya orqali isbotlangan ifodaga misol ko'rsating.
2. Yig'indi va Ko'paytmalarga 5 tadan misol ko'rsating.
3. Butun qiymatli funksiyalarni matematikada qollanishini ko'rsating.
4. O'rin almashtirishlar va faktoriallarni hisoblashga misol tuzing.
5. Fibonachi sonlaridan iborat 5 har hil ketma-ketlik tuzing.

### **Mavzuga doir testlar:**

1. Algoritm ma'lum bir ijrochiga muljallab tuziladi. Agar ijrochi EXM bulsa, algoritm kanday yozilishi kerak?
  - A) Blok sxemalar yordamida ifodalanishi kerak.
  - B) So'zlar yordamida yozilishi kerak.
  - C) So'zlar va formulalar yordamida
  - D) Jadval ko'rinishida ifodalanishi zarur.
2. Algoritm va EXM uchun dastur tushunchalari orasidagi fark nimadan iborat?
  - A) EXMga tushunarli tilda yozilgan algoritm dasturidir.
  - B) Ular bir xil tushunchalar
  - C) Xar kanday algoritm dastur bula oladi
  - D) Ular orasida hech kanday umumiylik yuk
3. Masalaning kuyilishidan nimalar aniklanadi?
  - A) Nima berilgan va nimani topish kerakligi
  - B) Algoritmning uzunligi
  - C) Dasturning bajarilish vakti;
  - D) Zaruriy xotira hajmi.

### **Adabiyotlar**

1. В.А.Успенский, А.Л.Семенов. Теория алгоритмов: основные открытия и приложения. – М: Наука, 1987, 287 с.
2. Т.Кормен, Ч.Лейзерсон, Р.Ривест. Алгоритмы: построение и анализ. Сер: Классические учебники. М.: МЦНМО, 2001.- 960 с.
3. Гуломов С.С. ва бошқалар. Ахборот тизимлари ва технологиялари. Тошкент, 2000 й.
4. Жуманов И.И. Мингбаев Н.С., Информатика.- Самарқанд,: СамДУ нашри, 2002, 107 бет.
5. Ahatov A.R., Zaripova G.L. va boshq. Axborot texnologiyalari // Uslubiy qo'llanma. – Samarqand: SamDU nashri, 2008 yil – 112 bet.
6. Д.Кнут. Искусство программирования для ЭВМ. Основные алгоритмы.-М: Мир, 2000 г.
7. Уотермен Д. Руководство по экспертным системам. М: Мир, 1989 г.
8. А.Ахо., Дж.Хопкрофт. Построение и анализ вычислительных алгоритмов. - М: Мир, 1979 г., 535 с.
9. Лебедев В.И. Введение в системы программирования. М: Статистика, 1975 г.

### 3 -4 MA'RUZALAR: ALGORITMLAR VA ULARNING TO'LIQ TUZULISHINING BOSQICHLARI

#### *Reja*

1. *Algoritmning ta'rifi.*
2. *Algoritmni to'liq yaratish bosqichlarni*
3. *Masalaning qo'yilishi.*
4. *Modelni yaratish.*
5. *Algoritmni ishlab chiqish.*
6. *Algoritm to'g'riligini tekshirish.*
7. *Algoritmni amalga oshirish.*
8. *Algoritmni va ularning murakkabligini tahlil qilish.*
9. *Dasturni tekshirish.*
10. *Hujjatlashtirish.*

**Darsning maqsadi:** talabalarga algoritm tushunchasi, ta'riflari, algoritmni tuzishning bosqichlari haqida ma'lumot berish.

**Tayanch iboralar:** algoritmlar nazariyasi, algoritmni tuzish, masala quyilishi, modellashtirish, test, ishlab chiqish, hujjatlashtirish.

**Mashg'ulot vositalari:** sinf doskasi, plakatlar, fundamental fan darsliklari, o'quv va uslubiy qo'llanmalar, informatika bo'yicha atamalar lug'ati, videoprojektor, ekran va kompyuterdan samarali foydalanish.

**Mashg'ulot usullari:** takrorlash, suhbat va savol-javob, munozara (mavzuni o'zlashtirishni mustahkamlash) tarzida muloqot o'tkazish, (talabalarning mustaqil, erkin fikrlash va so'zlashga o'rgatgan holda fikr mulohazalarini bayon qildirish, buning uchun har bir talabaga, tayanch iboralardan savollar tashlanadi, ular o'z fikrlarini bayon qiladilar, hamma talaba javobni bayon qilib bo'lgandan so'ng talaba bilan birgalikda javoblar yakun qilinadi).

**Darsning xronologik xaritasi** – 80 minut.

**Tashkiliy qismi:** Auditoriyaning jixozlanishi va sanitar sharoitlari, talabalar davomati – 2 minut.

**Bilimlarni baholash:** yangi mavzuni o'rganish uchun zarur bo'lgan material bo'yicha suxbat – 10 minut.

**Yangi mavzuni bayon etish** – 55 minut.

**Mavzu o'zlashtirilgan darajasini aniqlash** – 10 minut.

**Uyga vazifa** – 3 minut.

#### Algoritmning ta'rifi

Algoritmning turli ta'riflari mavjud. Rasmiy ta'riflardan biri bo'yicha algoritm bu qo'yilgan masalani bir xil yechilishiga olib keluvchi aniq harakatlarning ketma-ketligi. Bu tushunchadan algoritmning quyidagi xossalari kelib chiqadi:

5. Diskretlilik – ya'ni aniqlanayotgan jarayonni qadamba-qadam ko'rinishi.
6. Ommaviylik – algoritm o'xshash masalalar turkumini yechishi kerak.
7. Tushunarlilik – algoritmda beriladigan ko'rsatmalar foydalanuvchiga tushunarli bo'lib, uning talablariga javob berishi kerak.
8. Aniqlilik – algoritmda ma'lum tartibda amallarni bajarish nazarda tutilishi kerak va bajaruvchiga joriy qadam tugatilishi bilan qaysi qadam keyingi bo'lib bajarilishi aniq ko'rsatilishi kerak.

Algoritm rasm ravishda bajariladi, bu degani bajaruvchi bajarilayotgan amallarni mazmunini anglash shart emas. Algoritm tuzish jarayoniga algoritmlashtirish deyiladi.

### **Algoritmi to'liq yaratish bosqichlari**

Algoritm tuzish jarayonida nazariy va amaliy nuqtai nazardan algoritmlash, dasturlash va EHM larni qo'llash bilan bog'liq bo'lgan bilimlar kerak. Asosiy maqsad bu masalani qo'yish, masalaning yechish algoritmini tuzish, algoritmi mashina dasturi ko'rinishida amalga oshirish va algoritmi samaradorligini ko'rsatish muammolarini o'rganish. Bu jarayonlar algoritmi to'liq yaratish tushunchasiga olib keladi va quyidagi bosqichlarni belgilaydi:

9. Masalaning qo'yilishi.
10. Modelni yaratish.
11. Algoritmi ishlab chiqish.
12. Algoritm to'g'riligini tekshirish.
13. Algoritmi amalga oshirish.
14. Algoritmi va ularning murakkabligini tahlil qilish.
15. Dasturni tekshirish.
16. Hujjatlashtirish.

### **Masala qo'yilishi**

Masalani yechishdan oldin, uni berilishini aniq shakllantirib olish zarur. Bu jarayon to'g'ri savollarni aniqlash bo'lib, savollar quyidagicha bo'lishi mumkin:

- 1.7. Dastlabki berilgan masala shartlarida hamma iboralar tushunarlimi?
- 1.8. Nima berilgan?
- 1.9. Nimani topish kerak?
- 1.10. Yechimni qanday ta'riflash kerak?
- 1.11. Qaysi berilganlar yetarli emas va hammasi kerakmi?
- 1.12. Qanaqa mumkinliklar qabul qilingan?

Albatta, bulardan tashqari boshqa savollarni ham ishlatish mumkin, yoki ayrim savollarni bir necha bor takror ishlatishga to'g'ri keladi.

### **Modelni yaratish**

Akademik A. N. Tixonov fikri bo'yicha matematik modellashtirish dunyoni bilish va o'rganishda kuchli qurollardan (vositalardan) biridir.

Uning ta'rifi bo'yicha matematik model tashqi dunyoning xodisalar turkumini matematik belgilar yordamida taxminiy tavsifi.

Xodisani tavsiflash uchun uning muhim xususiyatlarini, qonuniyliklarini, ichki aloqalarini, ayrim xossalarning ahamiyatini aniqlash zarur. Eng muhim faktorlari aniqlanganda, ahamiyatlari kamroq bo'lganlarini hisobdan chiqarish mumkin. Umuman, modelni tanlash fandan ko'ra, ko'proq san'at ishi deb hisoblanadi, yahshi tuzilgan modellarni o'rganish esa – modellashtirishda tajriba orttirishning eng yahshi usuli. Modelni yaratishda quyidagi savollarni aniqlash maqsadga muvofiq:

- 2.7. Masalani yechish uchun qaysi matematik struktura ko'proq mos keladi?
- 2.8. O'xshash masalaning yechimi bormi?
- 2.9. Masalaning barcha muhim ma'lumotlari matematik ob'yektlar orqali tavsiflanadimi?
- 2.10. Izlanayotgan natija biron bir matematik o'lchamga mos keladimi?
- 2.11. Modelning ob'yektlari orasidagi bog'lanishlar aniqlanganmi?
- 2.12. Tuzilgan model bilan ishlash qulaymi?

### **Algoritmi ishlab chiqish**

Algoritmshatirish jarayoni uslublari bo'yicha matematik modellarni tuzish jarayoniga juda yaqin. Har bir algoritmi ishlab chiqish bevosita o'ziga xos yondashishni talab qilishiga qaramasdan, bu faoliyatni umumiy uslub va bosqichlari ham mavjud. Ba'zan dasturlarni tezroq yozib boshlashga hohish paydo bo'ladi. Lekin bu xatoli, chunki aynan algoritmi ishlab chiqish bosqichiga va uning to'g'riligiga masalaning to'liq yechimi bog'liqdir. Algoritmshatirish turli xil uslublari mavjud.

## Algoritmni to'g'riligini tekshirish

Dastur to'g'riligini isbotlashning eng keng tarqalgan turi – bu uni testlardan o'tkazishdir.

Algoritmni tekshirishda nazoratchi boshlang'ich ma'lumotlarni majmui algoritmik test deb nomlanadi.

To'g'ri deb shunday algoritmga aytiladiki, u masalaning qo'yilishida talab qilinadigan natijani har qanday ruxsat etilgan boshlang'ich ma'lumotlar bilan ham shakllantirib biladi. Odatda, dastur bergan natijalar ma'lum bo'lgan yoki qo'lda hisoblangan ma'lumotlar bilan taqqoslanadi, va ular to'g'riligi aniqlansa dastur to'g'ri ishlaydi degan hulosaga kelish mumkin. Ammo bu usul bilan foydalanuvchini hamma shubhalardan xalos qilib bo'lmaydi, ya'ni dastur ishlamaydigan hamma holatlarni hisobga olib bo'lmaydi.

Gudman va Xidetniyemi [2] lar tomonidan algoritm to'g'riligini isbotlash uchun quyidagi uslubiyat taklif qilingan.

Algoritm 0 dan  $m$  gacha bo'lgan qadamlar ketma-ketligi ko'rinishida tavsiflangan deb tahmin qilaylik. Har bir qadam uchun qandaydir asoslanishni taklif etamiz. Xususan, qadamdan oldin va keyin ishlaydigan shartlar haqida lemma kerak bo'lishi mumkin. Shu bilan birgalikda, algoritm chekliligining isbotini ham taklif etamiz, va hamma ruxsat etilgan kiritish ma'lumotlarini tekshirib, hamma mumkin bo'lgan chiqarish ma'lumotlarni olamiz. Algoritmni to'g'riligi bilan samaradorligi o'rtasida hech qanday aloqa yo'qligini ta'kidlab o'tamiz. Aslida hamma talablarga bir xil yahshi javob beradigan algoritm kamdan-kam ishlab chiqiladi.

## Algoritmni amalga oshirish

Algoritmni amalga oshirish deganda, EHM uchun dasturni yozish deb tushuniladi. Buning uchun quyidagi savollarga javob berish kerak:

- 5.7. Asosiy o'zgaruvchilarni aniqlash.
- 5.8. O'zgaruvchilarning turlarini aniqlash.
- 5.9. Nechta massiv yoki fayllar va qanday kattalikda ular kerak bo'ladi?
- 5.10. Bog'lanilgan ro'yhatlardan foydalanish ma'nolimi?
- 5.11. Qanday dasturiy qismlar kerak bo'lishi mumkin (tayyor bo'lsa ham)?
- 5.12. Qaysi dasturlash tilini tanlash?

Dastur yozish yoki tuzishning hilma-hil usillari va uslublari mavjud.

## Algoritmni va uning murakkabligini tahlil qilish

Algoritmni tahlil qilishdan maqsad – algoritmga ma'lumotlarni aniq muvaffaqiyatli qayta ishlash uchun kerak bo'ladigan xotira hajmi va ishlash vaqtining baholari va chegaralarini olish. Bir masalani yechadigan ikki algoritmni taqqoslash uchun qandaydirsonli mezon topish kerak.

Faraz qilaylik,  $A$  – qandaydir bir turkumdagi masalalarni yechadigan algoritm,  $n$  – esa shu turkumdagi alohida bir masalaning kattaligi. Umumiy holda,  $n$  – oddiy skalyar yoki massiv yoki kiritiladigan ketma – ketlikning uzunligi bo'lishi mumkin.  $f_A(n)$  -  $n$  kattalikdagi ixtiyoriy masalani yechadigan algoritm  $A$  bajarish kerak bo'lgan asosiy amallarni (qo'shish, ayirish, taqqoslash,...) yuqori chegarasini beradigan ishchi funksiya. Algoritmning sifatini baholash uchun quyidagi mezonni ishlatamiz.

Agar  $f_A(n)$  o'sish tartibi  $n$  dan bog'liq bo'lgan polinomdan katta bo'lmasa,  $A$  algoritm polinomial deb aytiladi, aks holda algoritm  $A$  eksponensial hisoblanadi.

Shular bilan birgalikda tahlil jarayonida ko'p matematik fanlarda standart bo'lgan iboralar ishlatiladi.

$f_A(n)$  funksiya  $O[g(n)]$  deb belgilanadi, va  $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = const \neq 0$  bo'lganda, uni tartibi

katta  $n$  lar uchun  $g(n)$  deb qabul qilinadi. Demak  $f(n) = O[g(n)]$ .

$f_A(n)$  funksiyasi  $o[z(n)]$  deb katta  $n$  lar uchun belgilanadi, va unda  $\lim_{n \rightarrow \infty} \frac{h(n)}{z(n)} = 0$  sharti

bajariladi.

Bu begilar “katta O” va “kichik o” deb nomlanadi. Agar  $f(n)=O[g(n)]$  bo’lsa, ikkala funksiya ham  $n \rightarrow \infty$  bo’lganda bir xil tezlikda o’sadi.

Agar  $f(n)=O[g(n)]$  bo’lsa, unda  $g(n)$ ,  $f(n)$  nisbatan ancha tez o’sadi.

Demak,  $P_k(n)$ - qandaydir  $n$  o’zgaruvchidan bog’liq va  $k$  darajadagi polinom uchun  $f_A(n) = O[P_k(n)]$  yoki  $f_A(n) = oP_k(n)$  bo’lganda algoritm polynomial hisoblanadi, aks holda algoritm eksponensial.

Eksponensial algoritm yahshi ishlamaydigan deb hisoblanadi. Agar algoritmlar eksponensial bo’lsa, ular orasida eng samaralisini topish kerak,  $n$  kattalikdagi masalani  $O(2^n)$  qadamda yechadigan algoritm  $O(n!)$  yoki  $O(n^n)$  qadamda masalani yechadigan algoritmdan afzalroq.

### **Dasturni tekshirish**

Biz dasturni har bir qismini tekshiradigan kirituvchi ma’lumotlar to’plamini tanlashimiz kerak. Ko’p murakkab algoritmlarni matematik tomondan tadqiq qilish yoki juda qiyin yoki mumkin emas. Bunday holatlarda algoritmni faoliyat jarayonida va qiyinligi bo’yicha tekshiradi. Bundan tashqari dasturlarni hisoblash imkoniyatlarini aniqlash uchun ham testlash maqsadga muvofiq. Ko’p dasturlar qandaydir kiritiladigan ma’lumotlar bilan yahshi ishlasa, boshqalari bilan yomon ishlaydi. “Yahshi” lardan “yomon” larga o’tish “mayin” bo’lish kerak. Testlash uchun ma’lumotlar dasturning qiyinligiga, mavjud vaqt resurslariga, kiritish-chiqarishsoniga bog’liq holda tanlanadi. Bu yerda analitik va eksperimental tahlil bir-birini to’ldiradi.

### **Hujjatlashtirish**

O’zingiz yozmagan dastur kodini o’qish juda qiyin. Bu muammoni hujjatlashtirish yordamida yechsa bo’ladi. Hujjatlashtirish o’z ichiga hamma yordamchi ma’lumotlarni oladi va dasturda nima bajarilishini tushuntirib beradi, xususan, blok-sxemalardagi boshqarishni uzatish, berilganlarni kiritish-chiqarish shaklini batafsil tavsif qilish, siklning parametrlari, yordamchi local va global proseduralarni bajarilishi va boshqalar.

Hujjatlashtirishning eng asosiy qoidasi bu “boshqalar yozgan dasturlarni qanday ko’rishni istasangiz, o’zingiz ham dasturni shunday ko’rinishda rasmiylashtiring”.

### **Takrorlash ucun savollar**

1. Algoritmning qaysi ta’riflarinin bilasiz?
2. Algoritmni to’liq yaratish bosqichlarini aytib o’ting
3. Masalani qo’yishda va modelni yaratishdagi savollarni qanday aniqlash kerak?
4. Algoritmni va ularning murakkabligini tahlil qilishda nimalarga e’tibor berish kerak?

### **Mustaqil ishlash uchun nazorat savollari:**

1. Algoritm ta’riflariga beshta misol ko’rsating.
2. Zamonaviy algoritmlarga nisbatan mavjudlaridan tashqari qanday hossalarni qo’shish mumkin?
3. Masala qo’yilishin aniqlashga va modelni yaratishga savollar sonini 10 ga yetkazing?
4. Algoritmni ishlab chiqishga amaliy misol va uning yechimini ko’rsating.
5. Biron-bir tuzilgan algoritm murakkabligini  $o$ - mezoni bilan baholang.
6. Dasturni tekshirish uchun kiritiladigan test ma’lumotlariga misol tuzing.
7. Ixtiyoriy matematik masala yechimiga bag’ishlangan algoritmni tanlang va uning qadamlariga izoh tuzib, hujjatlashtirish jarayonini amalgam oshiring.

### Mavzuga doir testlar:

1. Quyidagi bandlardan qaysi birida algoritm tushunchasi aniqroq va tuliqroq ta'riflangan?
  - A) Algoritm-quyilgan masalani yechish yoki ma'lum bir maksadga erishish uchun ijrochi bajarishi zarur bulgan ish xarakatning (amallarning) tushunarli va anik ketma-ketligidir.
  - B) Algoritm uzbek matematigi Al Xorazmiy nomi bilan boglik bulib, uning yevropacha buzib aytilishidir.
  - C) Algoritm deganda EXM uchun tuzilgan dasturni tushunamiz.
  - D) Algoritm ijrochiga berilgan kursatma (yuriknoma) bulib xizmat kiladi.
2. Quyidagi hossalardan qaysi biri bo'yicha algoritmning har bir qadami aniq belgilangan bo'lishi kerak?
  - A) Aniqlik
  - B) Ommaviylik
  - C) Moslashuvchanlik
  - D) Diskretlik
3. Algoritmning birinchi rasmiy tushunchasini kim kiritgan?
  - A) Gedel
  - B) Muxammad Al-Xorazmiy
  - C) Tyuring
  - D) Paskal
4. Algoritm tushunchasini intuitiv darajada birinchi bo'lib kim kiritgan?
  - A) Muxammad Al-Xorazmiy
  - B) Paskal
  - C) Tyuring
  - D) Gedel
5. Bugungi kunda eng keng tarqalgan algoritmning rasmiy ta'rifi muallifi kim?
  - A) Tyuring
  - B) Muxammad Al-Xorazmiy
  - C) Paskal
  - D) Fon-Neymanl
6. Muayan sinf masalalarini yechaydigan aniq belgilangan qadamlar ketma-ketligi bu ...
  - A) Algoritm
  - B) Qism-dastur
  - C) Massiv
  - D) Dastur
7. Algoritm ma'lum bir ijrochiga muljallab tuziladi. Agar ijrochi EXM bulsa, algoritm kanday yozilishi kerak?
  - A) Blok sxemalar yordamida ifodalanishi kerak.
  - B) So'zlar yordamida yozilishi kerak.
  - C) So'zlar va formulalar yordamida
  - D) Jadval ko'rinishida ifodalanishi zarur.
8. Algoritm va EXM uchun dastur tushunchalari orasidagi fark nimadan iborat?
  - A) EXMga tushunarli tilda yozilgan algoritm dasturdir.
  - B) Ular bir xil tushunchalar
  - C) Xar kanday algoritm dastur bula oladi
  - D) Ular orasida xech kanday umumiylik yuk

9. Algoritm yaratish jarayonining boskichlarini tugri tartibda joylashtiring:

- 9) Masalaning kuyilishi
- 10) Algoritmni yozish;
- 11) Model tuzish;
- 12) Algoritmni amalga oshirish (realizasiya);
- 13) Algoritm tugriligini tekshirish;
- 14) Dasturni tekshirish;
- 15) Algoritmni va uning murakkabligini taxlil kilish;
- 16) Xujjatlashtirish.
  - A) 1;3;2;5;4;7;6;8.
  - V) 1;2;3;4;5;6;7;8
  - S) 2;1;3;4;5;6;8;7
  - D) 1;4;2;5;3;8;7;6

10. Masalaning kuyilishidan nimalar aniklanadi?

- A) Nima berilgan va nimani topish kerakligi
- B) Algoritmning uzunligi
- C) Dasturning bajarilish vakti;
- D) Zaruriy xotira hajmi.

#### Adabiyotlar

1. В.А.Успенский, А.Л.Семенов. Теория алгоритмов: основные открытия и приложения. – М: Наука, 1987, 287 с.
2. Т..Кормен, Ч.Лейзерсон, Р.Ривест. Алгоритмы: построение и анализ. Сер: Классические учебники. М.: МЦНМО, 2001.- 960 с.
3. Гуломов С.С. ва бошқалар. Ахборот тизимлари ва технологиялари. Тошкент, 2000 й.
4. Жуманов И.И. Мингбаев Н.С., Информатика.- Самарқанд,: СамДУ нашри, 2002, 107 бет.
5. Ahatov A.R., Zaripova G.L. va boshq. Axborot texnologiyalari // Uslubiy qo'llanma. – Samarqand: SamDU nashri, 2008 yil – 112 bet.

#### 5 – MA'RUZA: ALGORITMLARNI TAVSIFLASH TILI HAQIDA KELISHUV

##### Reja

1. *Algoritmning umumiy ko'rinish*
2. *Tarmoqlanuvchi yoki shartli buyruqlar*
3. *Tanlash buyruqlari.*
4. *Takrorlash buyruqlari.*

**Darsning maqsadi:** talabalarga algoritmlarni tavsiflash tili tushunchasi, asosiy konstruksiyalar, ularni turli dasturlash tilidagi kurinish bo'yich ma'lumot berish.

**Tayanch iboralar:** algoritmlar nazariyasi, chiziqli, tarmoqlanuvchi, tanlash, takrorlanuvchi, modellashtirish, test, ishlab chiqish, hujaatlashtirish.

**Mashg'ulot vositalari:** sinf doskasi, plakatlar, fundamental fan darsliklari, o'quv va uslubiy qo'llanmalar, informatika bo'yicha atamalar lug'ati, videoprojektor, ekran va kompyuterdan samarali foydalanish.



**Mashg'ulot usullari:** takrorlash, suhbat va savol-javob, munozara (mavzuni o'zlashtirishni mustahkamlash) tarzida muloqot o'tkazish, (talabalarning mustaqil, erkin fikrlash va so'zlashga o'rgatgan holda fikr mulohazalarini bayon qildirish, buning uchun har bir talabaga, tayanch iboralardan savollar tashlanadi, ular o'z fikrlarini bayon qiladilar, hamma talaba javobni bayon qilib bo'lgandan so'ng talaba bilan birgalikda javoblar yakun qilinadi).

**Darsning xronologik xaritasi** – 80 minut.

**Tashkiliy qismi:** Auditoriyaning jixozlanishi va sanitar sharoitlari, talabalar davomati – 2 minut.

**Bilimlarni baholash:** yangi mavzuni o'rganish uchun zarur bo'lgan material bo'yicha suxbat – 10 minut.

**Yangi mavzuni bayon etish** – 55 minut.

**Mavzu o'zlashtirilgan darajasini aniqlash** – 10 minut.

**Uyga vazifa** – 3 minut.

Algoritmni tavsiflash usullari haqida aytadigan bo'lsak, ular so'zlar bilan, jadvallar bilan, grafik yoki blok-sxemalar va maxsus sun'iy tilda berilishi mumkin.

So'zli tavsiflashda tabiiy til va matematik belgilar elementlari ishlatiladi. Jadvalli berilishida algoritm jadvallar va hisoblash formulalari shaklida ko'rsatiladi. Grafikli, blok-sxemali va graflar yordamida berilishda algoritm geometrik figuralar va bloklar yordamida tavsiflanadi.

Algoritmik til – bu algoritm va ular bajarilishini bir qolirda va aniq yozish uchun belgilar va qoidalar tizimi. Algoritmik til so'zlarni tuzishga xizmat qiladigan o'z lug'atiga ega bo'ladi. So'zlar algoritmni buyruqlarini yozish uchun ishlatiladi. Bundan tashqari tilning xizmat so'zlari ham bo'ladi, ularning berilishi va ma'nosi aniq belgilangan.

So'zli algoritmik tillarga misol bo'lib "Informatika" kursidagi maktab algoritm tili xizmat qilishi mumkin. Bu til yordamida ixtiyoriy algoritmni o'qishva tahlilga qulay ko'rinishda, hamda unga qandaydir qo'shimcha buyruqlar va konstruksiyalar kiritish imkoni bilan yozish mumkin.

**Algoritmning umumiy ko'rinishi va shu tilning asosiy buyruqlari.**

**1. Algoritmning umumiy ko'rinishi:**

Alg nomi (argumentlar va natijalar ruyhati).

Arg ruyhat.

Nat ruyhat.

Bosh

Buyruqlar ruyhati.

Tamom.

**2. Tarmoqlanuvchi yoki shartli buyruqlar.**

Agar shart.

Unda ruyhat 1.

Aks holda ruyhat 2.

Tamom.

**3. Tanlash buyruqlari.**

Tanlash

Shart 1: ruyhat 1.

Shart 2: ruyhat 2.

.....

Shart N: ruyhat N.

Tamom.

**4. Takrorlash buyruqlari.**

1. toki shart.

Sikl bosh. ruyhat.

Sikl tug.

2. Takror ruyhat to shart .

3.  $i=n$  dan  $m$  gacha

sikl bosh ruyhat

sikl tug.

Bu tilda yozilgan algoritmlarni yuqori darajali dasturlash tiliga bevosita o'tkazish oson. Algoritmni tuzishda va tahlil qilishda bu yerda faqatgina qabul qilingan algoritmik tildagi konstruksiyaga mos buyruqlarni bajarish uchun kerak bo'ladigan vaqt va xotira muhim.

Takrorlash ucun savollar

1. Algoritmni tavsiflash uchun qaysi tillardan foydalansa bo'ladi?
2. Asosiy konstruksiyalarni blok-sxema yordamida ifodalang.
3. Asosiy konstruksiyalarni Paskal dasturlash tilida ifodalang.
4. Asosiy konstruksiyalarni C++ tilida ifodalang.

### Mustaqil ishlash uchun nazorat savollari:

1. Algoritm ta'riflariga beshta misol ko'rsating.
2. Algoritmni tavsiflash uchun tillarga 5 ta misol ko'rsating?
3. Chiziqli konstruksiyani blok-sxema yordamida ifodalang.
4. Tarmoqlanish konstruksiyalarni Paskal dasturlash tilida ifodalang.
5. Sharti oldin berilgan takrorlanuvchi konstruksiyalarni Paskal tilida ifodalang.
6. Sharti keyn berilgan takrorlanuvchi konstruksiyalarni Paskal tilida ifodalang.
7. Parametri berilgan takrorlanuvchi konstruksiyalarni Paskal tilida ifodalang.

### Mavzuga doir testlar:

1. Quyida

$$y = \frac{(x+1)^2 + 2(x+1)}{2(x+3)^2 + 3(x+3)}$$

ifodani xisoblash uchun algoritmlar keltirilgan. Ulardan qaysi biri eng samarali (effektiv) algoritm bo'la oladi?

A) Boshlanish

y:= x+1

y:=y^2+2y

b:=x+3

b:=2b^2+3b

y:=y/b

Tamom

B) Boshlanish

a:=x+1

b:=x+3

c:=a^2+2a

d:=2b^2+3b

y:=c/d

tamom

C) Boshlanish

$a:=x+1$

$a:=a^2+2a$

$b:=x+3$

$b:=2b^2+3b$

$y:=a/b$

Tamom

D) Boshlanish

$A:=x+1$

$B:=a^2$

$C:=x+3$

$D:=c^2$

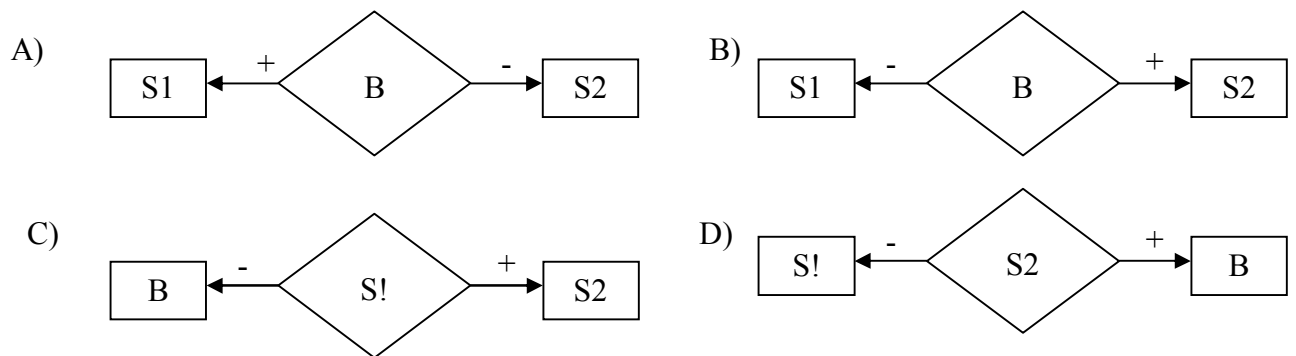
$E:=b+2a$

$F:=2d+3c$

$Y:=e/f$

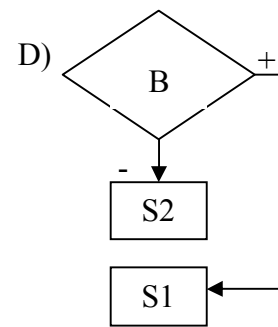
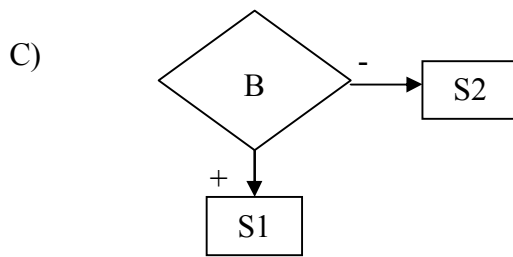
Tamom

2. Shartli operator If B then S1 else S2 ning bajarilishiga mos blok-sxemani kursating.

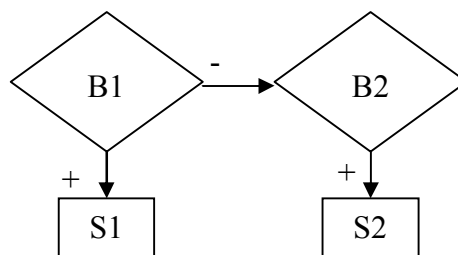


3. Quyidagi If B then S1; S2; operatorlarning bajarilishiga mos blok-sxemani kursating.





52. Quyidagi



(bu yerda V1 va V2- mantiqiy ifodalar, S1,S2-operatorlar) blok-sxemaga mos shartli operatori kursating:

- A) If B1 then S1 else if B2 then S2;
- B) If B1 then if B2 then S1 else S2;
- C) If B1 then Begin if B2 then S1 End else S2;
- D) If B1 then S1 else Begin if B2 then S2 End;

#### Adabiyotlar

1. В.А.Успенский, А.Л.Семенов. Теория алгоритмов: основные открытия и приложения. – М: Наука, 1987, 287 с.
2. Т..Кормен, Ч.Лейзерсон, Р.Ривест. Алгоритмы: построение и анализ. Сер: Классические учебники. М.: МЦНМО, 2001.- 960 с.
3. Лебедев В.И. Введение в системы программирования. М: Статистика, 1975 г.
4. Интеллектуализация ЭВМ. Перспективы развития вычислительной техники. Под ред. Ю.М.Смирнова. М: 1989 г.
5. Мингбаев Н.С., Жуманов И.И. Информатика.- Самарканд,: СамДУ нашри, 2002, 107 бет.

## 6 – MA’RUZA: ALGORITMLAR VA ULARNING MURAKKABLIGI TUSHUNCHASI

### *Reja*

- 1. Algoritmni baholash mezonlari.**
- 2. Algoritmni vaqt qiyinligi bo'yicha optimallashtirish.**
- 3. Algoritmni hajmiy qiyinligi bo'yicha optimallashtirish.**

**Darsning maqsadi:** talabalarga algoritmlarni murakkabligini baholash, mavjud mezonlar baholash uslubiyati, asosiy funktsiyalari haqida ma'lumot berish.

**Tayanch iboralar:** algoritmlar nazariyasi, murakkablik, vaqtli, hajmiy, mezon, chegara, modellashtirish, optimallashtirish, test, ishlab chiqish, hujaatlashtirish.

**Mashg'ulot vositalari:** sinf doskasi, plakatlar, fundamental fan darsliklari, o'quv va uslubiy qo'llanmalar, informatika bo'yicha atamalar lug'ati, videoprojektor, ekran va kompyuterdan samarali foydalanish.

**Mashg'ulot usullari:** takrorlash, suhbat va savol-javob, munozara (mavzuni o'zlashtirishni mustahkamlash) tarzida muloqot o'tkazish, (talabalarning mustaqil, erkin fikrlash va so'zlashga o'rgatgan holda fikr mulohazalarini bayon qildirish, buning uchun har bir talabaga, tayanch iboralardan savollar tashlanadi, ular o'z fikrlarini bayon qiladilar, hamma talaba javobni bayon qilib bo'lgandan so'ng talaba bilan birgalikda javoblar yakun qilinadi).

**Darsning xronologik xaritasi** – 80 minut.

**Tashkiliy qismi:** Auditoriyaning jixozlanishi va sanitar sharoitlari, talabalar davomati – 2 minut.

**Bilimlarni baholash:** yangi mavzuni o'rganish uchun zarur bo'lgan material bo'yicha suxbat – 10 minut.

**Yangi mavzuni bayon etish** – 55 minut.

**Mavzu o'zlashtirilgan darajasini aniqlash** – 10 minut.

**Uyga vazifa** – 3 minut.

Algoritmlarni baholash uchun ko'pgina mezonlar mavjud. Odatda kirituvchi berilganlarni ko'payishida masalani yechish uchun kerak bo'ladigan vaqt va xotira hajmlarini o'sish tartibini aniqlash muammosi qo'yiladi. Har bir aniq masala bilan kiritiladigan berilganlarni miqdorini aniqlovchi qandaydir sonni bog'lash zarur. Bunday son masalaning kattaligi deb qabul qilinadi. Masalan, ikkita matritsani ko'paytirish masalasining o'lchami bo'lib, matritsalar kattaligiga xizmat qilishi mumkin. Graflar haqidagi masalada o'lcham sifatida graf shohlarining soni bo'lishi mumkin.

Algoritm sarflanayotgan vaqt masalaning o'lchami funksiyasi sifatida algoritmni vaqt bo'yicha qiyinligi deb nomlanadi. Bunday funktsiyaga masalaning kattaligi oshganda limit ostidagi o'zgarish asimptotik qiyinlik deb aytiladi.

Shunga o'xshab, hajmiy qiyinlik va asimptotik hajmiy qiyinlikni aniqlash mumkin.

Aynan algoritmning asimptotik qiyinligi natijada shu algoritm yordamida yechiladigan masalarni kattaligini aniqlaydi. Agar algoritm  $n$  kattalikdagi kirishlarni  $C \cdot n^2$  vaqtda qayta ishlasa ( $c$ -const), unda algoritmning vaqt bo'yicha qiyinligi  $O(n^2)$  teng deb hisoblanadi, va  $n$  tartibda deb aytiladi.

Hisoblash mashinalar tezligi oshishiga qaramasdan, ular yordamida yechilayotgan masalalar kattaligini oshishini algoritm qiyinligini tahlil orqali aniqlaydi.

Faraz qilaylik,  $A1, A2, \dots, A5$  nomli 5 ta algoritm quyidagi vaqtli qiyinliklar bilan berilgan.

Algoritm	Vaqtli qiyinlik
A1	$N$
A2	$N \log_2 n$
A3	$N^2$
A4	$N^3$
A5	$2^n$

Bu yerda vaqtli qiyinlik – bu  $n$  kattalikdagi kirishlarni qayta ishlash uchun kerak bo'ladigan vaqt birliklar soni. Masalan, vaqt birligini 1 millisekund deb qabul qilaylik.

Bunda A1 algoritm bir sekunda 1000 kattalikdagi kirishni qayta ishlash mumkin, A5 algoritmi esa kirish kattaligidagina 9 dan oshirib bilmaydi.

Keyingi jadval 1 sekunda, 1 minutda, 1 soatda 5 ta algoritmlarni har birining yordamida yechiladigan masalaning kattaligi keltirilgan.

Algoritm	Vaqtli qiyinlik	Masalaning maksimal o'lchami		
		1 sek	1 min	1 soat
A1	$N$	1000	60*100	$3,6 * 10^6$
A2	$N \log_2 n$	140	4893	$2 * 10^4$
A3	$N^2$	31	244	1897
A4	$N^3$	10	39	153
A5	$2^n$	9	15	21

Faraz qilaylik, keyingi avlod hisoblash mashinalari birinchi jadvalga nisbatano'n barobar tezligi oshadi. Keyingi jadvalda shunday oshishga nisbatan yechiladigan masalalar kattaligining oshishi ko'rsatilgan.

Algoritm	Vaqtli qiyinlik	Masalaning maksimal kattaligi		
		1 sek	1 min	1 soat
A1	$N$	$s_1$	$10s_1$	$\approx 10$
A2	$N \log_2 n$	$s_2$	$10s_2$	$\approx 10$
A3	$N^2$	$s_3$	$3,16s_3$	$\approx 3$
A4	$N^3$	$s_4$	$2,15s_4$	$\approx 2$
A5	$2^n$	$s_5$	$s_5 + 3,3$	$\approx 10/3$

Bu yerda A5 algoritmi uchun tezlikni 10 barobar oshishi masalaning kattaligining uchga oshishiga olib keladi. A3 algoritmi esa kattalik uch barobardan ziyod oshadi. Endi, tezlik oshishining o'rniga algoritmi kiruvchi berilganlarning hajmini oshishini ko'ramiz. Birinchi jadval bo'yicha taqqoslash asosi sifatida 1 min. ni qabul qilsak, A4 algoritmi o'rniga A3 ni qo'llaganimizda, masalaning kattaligi 6 barobar oshadi, A4 algoritmi o'rniga A2 ni qo'llaganda esa 125 barobar oshirilishga erishamiz. Agar taqqoslash asosi sifatida 1 soatni qabul qilsak, natijalar yanada ham muhimlashadi.

Algoritmi va uning qiyinligini batafsilroq muhokama qilish uchun biz algoritmi amalga oshirish uchun qo'llaniladigan hisoblash qurilmalarning modelini va hisoblashning elementar qadamini aniqlashimiz zarur. Afsus-ki, sharoitlarga mos keladigan modelni o'zi yo'q. Eng katta qiyinchilikni mashina so'zlarining kattaliklari tug'diradi. Masalan, agar mashina so'zi ixtiyoriy uzunlikda butun son shaklini qabul qilsa, unda butun masalaning kodi mashina so'zi ko'rinishdagi bir son bo'lishi mumkin. Lekin mashina so'zining uzunligi cheklangan bo'lsa, unda masalaning kattaligi kamroq bo'lganda muammolar yechilsa ham, oddiy katta sonlarni xotiralashda qiyinchiliklar tug'ilishi mumkin.

### **Takrorlash uchun nazorat savollari**

- 1. Algoritmi baholash mezonlari nima bilan farqlanadi?*
- 2. Algoritmi vaqt qiyinligini qanday hisoblash kerak?*
- 3. Algoritmi qaysi mezon bo'yicha optimallashtirish samarali?*

### **Mustaqil ishlash uchun nazorat savollari:**

1. Algoritmi baholash uchun qo'llanishi mumkin bo'lgan mezonlarni tavsiflab bering.
2. Vaqtli mezon bo'yicha baholash jarayoniga misollar ko'rsating.
3. Hajmiy mezon bo'yicha baholash jarayoniga misollar ko'rsating.
4. Asimptotik baholash mohiyatini tushuntirib bering.
5. Eksponentsial algoritmlarga misollar ko'rsating.
6. Polinomial algoritmlarga misollar ko'rsating.

### **Mavzuga doir testlar:**

1. Quyidagi bamlardan kaysi birida algoritmi tushunchasi anikrok va tulikrok ta'riflangan?
  - A) Algoritmi-quyilgan masalani yechish yoki ma'lum bir maksadga erishish uchun ijrochi bajarishi zarur bulgan ish xarakatning (amallarning) tushunarli va anik ketma-ketligidir.
  - B) Algoritmi uzbek matematigi Al Xorazmiy nomi bilan boglik bulib, uning yevropacha buzib aytilishidir.
  - C) Algoritmi deganda EXM uchun tuzilgan dasturni tushunamiz.
  - D) Algoritmi ijrochiga berilgan kursatma (yuriknoma) bulib xizmat kiladi.
2. Algoritmi va EXM uchun dastur tushunchalari orasidagi farq nimadan iborat?
  - A) EXMga tushunarli tilda yozilgan algoritmi dasturdir.
  - B) Ular bir xil tushunchalar
  - C) Xar kandy algoritmi dastur bula oladi
  - D) Ular orasida xech kandy umumiylik yuk

3. Algoritmning samaradorligini baholash uchun mezonlar:

- A) xotira hajmi va ijro vakti;
- B) aniqlik va tushunarliklik;
- C) zaruriy xotira hajmi;
- D) tug'rilik va aniqlik

4. Algoritmni tugri deymiz, agar

- A) u quyilgan masalaga mos yechimni bersa;
- B) u albatta sonli yechim bersa;
- C) u oxirigacha ishlasa;
- D) u xatolardan xoli bulsa.

5. Algoritmni aniq deymiz, agar

- A) Uning barcha kadamlari anik bulib, ularni boshkacha talkin kilish mumkin bulmasa;
- B) Uning barcha kadamlari sonli natijaga olib kelsa;
- C) Unda matematik model tugri bulsa;
- D) Xotira hajmi eng kam mikdorda bulsa.

#### **Adabiyotlar**

1. В.А.Успенский, А.Л.Семенов. Теория алгоритмов: основные открытия и приложения. – М: Наука, 1987, 287 с.
2. Т..Кормен, Ч.Лейзерсон, Р.Ривест. Алгоритмы: построение и анализ. Сер: Классические учебники. М.: МЦНМО, 2001.- 960 с.
3. Гуломов С.С. ва бошқалар. Ахборот тизимлари ва технологиялари. Тошкент, 2000 й.
4. Жуманов И.И. Мингбаев Н.С., Информатика.- Самарқанд,: СамДУ нашри, 2002, 107 бет.
5. Ahatov A.R., Zaripova G.L. va boshq. Axborot texnologiyalari // Uslubiy qo'llanma. – Samarqand: SamDU nashri, 2008 yil – 112 bet.
6. Интеллектуализация ЭВМ. Перспективы развития вычислительной техники. Под ред. Ю.М.Смирнова. М: 1989 г.
7. Тыгу Х. Концептуальное программирование. М: Наука, 1984.
8. Н. Вирт. Алгоритмы и структуры данных. – Досса, Хамарайан, 1997.
9. Жуманов И.И., Мингбоев Н.С. Ҳисоблаш системаларининг информацион асослари. Самарқанд,: СамДУ нашри, 2002, 107 бет.



## 7 MA'RUZA: ALGORITMLARNI ISHLAB CHIQISH METODLARI. MAKSIMUM TOPISH MASALASI.

### *Reja*

1. *Masalaning qo'yilishi.*
2. *So'zli algoritmni ishlab chiqish*
3. *Algoritmni tahlil qilish*

**Darsning maqsadi:** talabalarga algoritmlarni ishlab chiqish usullari haqida ma'lumot berish. Aniq misolda algoritmni ishlab chiqish va uni optimallashtirish bo'yicha ma'lumot berish

**Tayanch iboralar:** algoritmlar nazariyasi, minimum, maksimum, murakkablik, vaqtli, hajmiy, mezon, chegara, optimallashtirish, test, ishlab chiqish, hujaatlashtirish.

**Mashg'ulot vositalari:** sinf doskasi, plakatlar, fundamental fan darsliklari, o'quv va uslubiy qo'llanmalar, informatika bo'yicha atamalar lug'ati, videoprojektor, ekran va kompyuterdan samarali foydalanish.

**Mashg'ulot usullari:** takrorlash, suhbat va savol-javob, munozara (mavzuni o'zlashtirishni mustahkamlash) tarzida muloqot o'tkazish, (talabalarning mustaqil, erkin fikrlash va so'zlashga o'rgatgan holda fikr mulohazalarini bayon qildirish, buning uchun har bir talabaga, tayanch iboralardan savollar tashlanadi, ular o'z fikrlarini bayon qiladilar, hamma talaba javobni bayon qilib bo'lgandan so'ng talaba bilan birgalikda javoblar yakun qilinadi).

**Darsning xronologik xaritasi** – 80 minut.

**Tashkiliy qismi:** Auditoriyaning jixozlanishi va sanitar sharoitlari, talabalar davomati – 2 minut.

**Bilimlarni baholash:** yangi mavzuni o'rganish uchun zarur bo'lgan material bo'yicha suxbat – 10 minut.

**Yangi mavzuni bayon etish** – 55 minut.

**Mavzu o'zlashtirilgan darajasini aniqlash** – 10 minut.

**Uyga vazifa** – 3 minut.

Yuqorida orttirilgan bilimlar yordamida bir tipik masalani yechamiz:

### **Masalaning qo'yilishi.**

$x_1, x_2, \dots, x_n$  berilgan elementlar bo'yicha  $m$  va  $j$  larni shunday topingki,  $m = \max_{x_k} \{1 \leq k \leq n\} = x_j$  bo'lsin. Bu yerda  $j$  mumkin bo'lgancha maksimal bo'lsin.

### **So'zli algoritm**

8. Boshlanish.
9.  $j := n; k := n - 1; m := xn;$
10. agar  $k := 0$  unda 7 o'ting.
11. agar  $xk \leq m$  unda 6 o'ting.
12.  $j := k; m := xk;$
13.  $k := k - 1; 3$  o'ting;
14. tamom.

Algoritm sodda va analizga muhtoj emas deb hisoblanadi. Lekin shu misolda murakkab algoritmn qanday tahlil qilish kerakligini ko'rsatish mumkin. Algoritm tahlili dasturlash uchun juda muhim.

Biz faqatgina bu algoritmni bajarish uchun kerak bo'ladigan vaqtni tahlil qilamiz. Buning uchun har bir qadam necha marta bajarilishini hisoblaymiz:

Qadam raqami	Necha marta bajarilishi
2	1
3	n
4	n-1
5	A
6	n-1

Har bir qadam necha marta bajarilishini bilgan holda, kompyuterga masalani bajarish uchun qancha vaqt kerakligini hisoblab chiqish mumkin.

Jadvalda A dan tashqari hamma qiymatlar ma'lum, A – bu joriy maksimum qiymatini necha marta o'zgartirish kerakligini ko'rsatkichi. Taxlilimiz to'liq bo'lishi uchun A ni ko'rib chiqamiz.

Tahlilning maqsadi A uchun min va max qiymatlarni topish.

$$1) \text{Min } A = 0,$$

bu holat

$$x_n = \max_{k \in \{1 \leq k \leq n\}} x_k$$

bo'lganda kuzatiladi.

$$2) \text{Max } A = n-1;$$

bu qiymatga

$$x_1 > x_2 > \dots > x_n$$

holatida erishiladi.

Shunday qilib A ning tahlili 0 va n-1 larning o'rta arifmetik qiymati va o'rta kvadratini chetlanishini va usullari yordamida topish masalasiga olib keladi.

### Takrorlash ucun nazorat savollar

1. Masala quyilishida qaysi o'zgaruvchilar aniqlandi?
2. Algoritm qanday konstruksiyalar qatnashgan?
3. Aniqlangan noma'lum qiymat nechanchi qadamda bajariladi?
4. Algoritm tahlilini yakunga yetkazish ucun qanday usullarni qo'llash kerak?

### Mustaqil ishlash uchun nazorat savollari:

1. Algoritmni baholash uchun qo'llanishi mumkin bo'lgan mezonlarni tavsiflab bering.
2. Vaqti mezon bo'yicha baholash jarayoniga misollar ko'rsating.
3. Hajmiy mezon bo'yicha baholash jarayoniga misollar ko'rsating.
4. Minimum topish yechimini beradigan masalalarga 5ta misol ko'rsating
5. Maximum topish yechimini beradigan masalalarga 5ta misol ko'rsating

### **Mavzuga doir testlar:**

1. Algoritmning samaradorligini baholash uchun mezonlar:

- A) xotira hajmi va ijro vahti;
- B) aniqlik va tushunarliklik;
- C) zaruriy xotira hajmi;
- D) tug'rilik va aniqlik

2. Algoritmni tugri deymiz, agar

- A) u quyilgan masalaga mos yechimni bersa;
- B) u albatta sonli yechim bersa;
- C) u oxirigacha ishlasa;
- D) u xatolardan xoli bulsa.

3. Algoritmni aniq deymiz, agar

- A) Uning barcha kadamlari anik bulib, ularni boshkacha talkin kilish mumkin bulmasa;
- B) Uning barcha kadamlari sonli natijaga olib kelsa;
- C) Unda matematik model tugri bulsa;
- D) Xotira hajmi eng kam mikdorda bulsa.

### **Adabiyotlar**

1. В.А.Успенский, А.Л.Семенов. Теория алгоритмов: основные открытия и приложения. – М: Наука, 1987, 287 с.
2. Т..Кормен, Ч.Лейзерсон, Р.Ривест. Алгоритмы: построение и анализ. Сер: Классические учебники. М.: МЦНМО, 2001.- 960 с.
3. Гуломов С.С. ва бошқалар. Ахборот тизимлари ва технологиялари. Тошкент, 2000 й.
4. Жуманов И.И. Мингбаев Н.С., Информатика.- Самарқанд,: СамДУ нашри, 2002, 107 бет.
5. Ahatov A.R., Zaripova G.L. va boshq. Axborot texnologiyalari // Uslubiy qo'llanma. – Samarqand: SamDU nashri, 2008 yil – 112 bet.
6. Интеллектуализация ЭВМ. Перспективы развития вычислительной техники. Под ред. Ю.М.Смирнова. М: 1989 г.
7. Тыгу Х. Концептуальное программирование. М: Наука, 1984.
8. Н. Вирт. Алгоритмы и структуры данных. – Досса, Хамарайан, 1997.
9. Жуманов И.И., Мингбоев Н.С. Ҳисоблаш системаларининг информацион асослари. Самарқанд,: СамДУ нашри, 2002, 107 бет.

## 8 MA'RUZA: EVKLID ALGORITMINING TAHLILI

### *Reja*

- 1. Masala qo'yilishi.*
- 2. Algoritmni tuzish*
- 3. Algoritm tahlili*
- 4. Algoritm optimallashtirish*
- 5. Algoritmni amalga oshirish*

**Darsning maqsadi:** talabalarga algoritmlarni ishlab chiqishni Evklid algoritmi misolida ko'rsatish Aniq misolda algoritmni baholash va optimallashtirish bo'yicha ma'lumot berish

**Tayanch iboralar:** algoritmlar nazariyasi, minimum, maksimum, murakkablik, vaqti, hajmiy, mezon, chegara, optimallashtirish, test, ishlab chiqish, hujaatlashtirish.

**Mashg'ulot vositalari:** sinf doskasi, plakatlar, fundamental fan darsliklari, o'quv va uslubiy qo'llanmalar, informatika bo'yicha atamalar lug'ati, videoprojektor, ekran va kompyuterdan samarali foydalanish.

**Mashg'ulot usullari:** takrorlash, suhbat va savol-javob, munozara (mavzuni o'zlashtirishni mustahkamlash) tarzida muloqot o'tkazish, (talabalarning mustaqil, erkin fikrlash va so'zlashga o'rgatgan holda fikr mulohazalarini bayon qildirish, buning uchun har bir talabaga, tayanch iboralardan savollar tashlanadi, ular o'z fikrlarini bayon qiladilar, hamma talaba javobni bayon qilib bo'lgandan so'ng talaba bilan birgalikda javoblar yakun qilinadi).

**Darsning xronologik xaritasi** – 80 minut.

**Tashkiliy qismi:** Auditoriyaning jixozlanishi va sanitar sharoitlari, talabalar davomati – 2 minut.

**Bilimlarni baholash:** yangi mavzuni o'rganish uchun zarur bo'lgan material bo'yicha suxbat – 10 minut.

**Yangi mavzuni bayon etish** – 55 minut.

**Mavzu o'zlashtirilgan darajasini aniqlash** – 10 minut.

**Uyga vazifa** – 3 minut.

### Masala qo'yilishi

Ikkita butun musbat  $m$  va  $n$  sonlar berilgan. Ularning umumiy bo'luvchisini topish talab qilinadi. Ya'ni, eng katta butun musbat son topish kerakki, unga  $m$  va  $n$  ni bo'lganda butun son chiqsin.

### Algoritmni tuzish

6. Boshlash:
- $m$  ni  $n$  ga bo'lamiz, qoldiq  $r$  ga teng bo'lsin;
8. Agar  $r=0$  unda  $n$ -natija; 5 o'ting;
- $m:=n$ ;  $n:=r$ ; 2 o'ting;
10. tamom.

### Algoritm tahlili

Shu algoritmni tadqiq qilib ko'raylik.  $m=119$ ,  $n=544$  deb qabul qilaylik. Ikkinchi qadamdan boshlaymiz. Algoritmga binoan bo'lish natijasini nolga teng deb hisoblaymiz va  $r$  ga 119 ni ta'minlaymiz, keyin 3-qadamga o'tamiz.  $R$  nolga teng bo'lmaganligi uchun, hech nima qilmaymiz va 4-qadamga o'tamiz. Bu yerda  $m$  ga 544 ni,  $n$  ga 119 ni ta'minlaymiz. Umuman, ravshan bo'ldiki,  $m < n$  bo'lsa, 2-qadamda  $m$  va  $n$  larga nisbatan hech qanday amallar bajarilmaydi, algoritm esa  $m$  va  $n$  o'zgaruvchilar qiymatlari o'rin almashishiga olib keladi.

### **Algoritm optimallashtirish**

Algoritmni optimallashtirish uchun unga quyidagi qadamni qo'shamiz:

1.2. agar  $m < n$   $t := n$ ;  $n := m$ ;  $m := t$ ;

Endi 2-qadamga kelsak,  $544:119=4,68/119$ .  $r$  ga 68 ni ta'minlaymiz. 3-qadam ishlaymaydi. 4-qadamda  $n=68$ ,  $m=544$ ,  $r=68$ . Keyingi sikllarda ( $r=51$ ,  $m=68$ ,  $n=51$ ), keyin ( $r=17$ ,  $m=51$ ,  $n=17$ ) va  $51/17$ , ya'ni  $r=0$ .

Shunday qilib, algoritm sikli 3- qadamda tugadi va 544 va 119 larning umumiy bo'luvchisi 17 ga teng bo'ldi.

Bu algoritm umumiy bo'luvchini topish uchun yagona emas. Bunday algoritmni topish uchun Dj. Steynning binar algoritmi, yoki V, xorrisning algoritmidan foydalaniladi.

### **Algoritmni amalga oshirish**

Shu algoritmni kompyuterda amalga oshirish uchun quyidagi Paskal dasturini keltirish mumkin:

```
Program evklid;
Var a, b, nod, r, x, y: integer;
Begin read (a, b);
if a>b then begin x:=a; y:=b; end;
else begin x:=b; b:=a; end;
if (x>0) and (y>=0) then begin while y<>0 do
begin r:=x mod y; x:=y; y:=r; end;
Nod:=x; write (nod);
end; else write ('berilganlarda xato');
end.
```

### **Takrorlash ucun nazorat savollari**

1. Masala qo'yilishidagi o'zgaruvchilarni aniqlang.
2. Algoritm optimallashtirish uchun qanday qadam qo'shildi?
3. Algoritm qaysi dasturlash tilida amalga oshirildi?

### **Mustaqil ishlash uchun nazorat savollari:**

1. Algoritmni baholash uchun qo'llanishi mumkin bo'lgan mezonlarni tavsiflab bering.
2. Vaqtli mezon bo'yicha baholash jarayoniga misollar ko'rsating.
3. Hajmiy mezon bo'yicha baholash jarayoniga misollar ko'rsating.
4. Evklid algoritmi yordamida yechiladigan masalalarga 5ta misol ko'rsating
5. Evklid algoritmini turli dastirlash tillarida amalgam oshirib tahlil qiling

### **Mavzuga doir testlar:**

1. Quyidagi algoritmda siklning operatorlari necha marta bajariladi?

```
m:=36; n:=56
while m<>n do
if m>n then m:=m-n
else n:=n-m;
```

- A) 6
- B) 4
- C) 1
- D) 8

2. Agar o'zgaruvchilar tavsiflanishi

```
Type room=1..30;
Var x: real; y: byte; z: room;
```

bo'lsa, xatosiz bajarilayotgan buyruşlarni toping.

- A) Z: =30
- B) Z: =x
- C) x: =12; z:=x;
- D) X=y; z: =x

3. X va U uzgaruvchilarning dastlabki qiymatlari mos ravishda 0.9 va -1.5. Quyidagi shartli operator IF X<Y THEN X:=Y ELSE Y:=X; bajarilgandan sung ularning kiymati nimaga teng buladi?

- A) X=0.9 ; Y=0.9
- B) X=0.9 ; Y=-1.5
- C) X=-1.5; Y=0.9
- D) X=-1.5; Y=-1.5

4. Quyidagi

$$N = \begin{cases} 1, \text{ agar } y \geq 0 \text{ va } x \geq 0 \\ 2, \text{ agar } y \geq 0 \text{ va } x < 0 \\ 3, \text{ agar } y < 0 \text{ va } x < 0 \\ 4, \text{ agar } y < 0 \text{ va } x \geq 0 \end{cases}$$

ifoda kiymatini xisoblash uchun keltirilgan shartli operatorlardan kaysi biri tugri?

- A) Keltirilgan operatorlardan xech biri berilgan ifodani tugri xisoblamaydi.
- B) If y<0 then Begin x<0 then N:=3 else N:=4End  
else If x<0 then N:=2 else N:=1;
- C) If (y>=0) and (x>=0) then N:=1 else N:=2;  
If (y<0) and (x<0) then N:=3 else N:=4;
- D) If x<0 then Begin y<0 then N:=1 else N:=2 End  
else Begin If y>=0 then N:=3 else N:=4 End ;

5. Ta'minlash operatori kaday ishni bajarish uchun muljallangan? Eng umumiy javobni toping.

A) Operatorning ung kismida turgan ifodani xisoblaydi va uning kiymatini chap kismdagi uzgaruvchiga ta'minlaydi.

V) Uzgaruvchilarga kiymat ta'minlaydi.

S) Uzgaruvchilarning turini boshkasiga uzgartiradi.

D) Ifoda qiymati qaysi turga mansubligini aniqlaydi.

6. Quyidagi sanoq skalyar turlarni tavsiflash va ularga tegishli uzgaruvchilar ustida amallar bajarishga doir misollar keltirilgan. Bu misollardan kaysi biri xatosiz yozilgan?

A) Type T1=(AMAD, CAMAD, BYRI, ALI);

T2=(OQ, QORA, KUK, KIZIL);

VAR X,Y:T1;A,B:T2;

X:ALI; A:=KUK; B:=OQ

B) Type T1=(MEN, CEN, Y, 0.5);

T2=(INB, FEV, MART, APR, MAI, JUH);

C) Type T1 =(KATTA, KICHIK, URTA);

T2=(STOL,STUL,PARTA);

VAR X,Y,:T1;A,B:T2;

X:STOL;Y:=KICHIK;T2:=URTA;

D) Type T1 =(KATTA, KICHIK, URTA);  
T2=(STOL,STUL,PARTA);  
VAR X,Y: boolean;  
X:=KATTA<KICHIK;  
Y:=STUL>URTA;

#### Adabiyotlar

1. В.А.Успенский, А.Л.Семенов. Теория алгоритмов: основные открытия и приложения. – М: Наука, 1987, 287 с.
2. Т.Кормен, Ч.Лейзерсон, Р.Ривест. Алгоритмы: построение и анализ. Сер: Классические учебники. М.: МЦНМО, 2001.- 960 с.
3. Гуломов С.С. ва бошқалар. Ахборот тизимлари ва технологиялари. Тошкент, 2000 й.
4. Жуманов И.И. Мингбаев Н.С., Информатика.- Самарқанд,: СамДУ нашри, 2002, 107 бет.
5. Ahatov A.R., Zaripova G.L. va boshq. Axborot texnologiyalari // Uslubiy qo'llanma. – Samarqand: SamDU nashri, 2008 yil – 112 bet.
6. Интеллектуализация ЭВМ. Перспективы развития вычислительной техники. Под ред. Ю.М.Смирнова. М: 1989 г.
7. Тыгу Х. Концептуальное программирование. М: Наука, 1984.
8. Н. Вирт. Алгоритмы и структуры данных. – Досса, Хамарайан, 1997.
9. Жуманов И.И., Мингбоев Н.С. Ҳисоблаш системаларининг инфорацион асослари. Самарқанд: СамДУ нашри, 2002, 107 бет.

### 9 -10 MA'RUZA: KOMMIVOYAJER MASALASINI ECHISH USLUBLARI. EVRISTIK ALGORITMLAR ASOSIDA MASALALARNI YECHISH

#### REJA

1. *Masala qo'yilishi.*
2. *Evristik algoritmlar.*
3. *GTS algoritmini tuzish*
4. *Algoritmni baholash*

**Darsning maqsadi:** talabalarga algoritmlarni ishlab chiqishning evristik usullari haqida tushuncha berish. Kommivoyajer masalasi yordamida aniq algoritmgga misol ko'rsatish  
Algoritmni baholash va optimallashtirish bo'yicha kunikma hosil qilish

**Tayanch iboralar:** algoritmlar nazariyasi, evrisika, marshrut, minimum, maksimum, murakkablik, vaqqli, hajmiy, mezon, chegara, optimallashtirish, test, hujaatlashtirish.

**Mashg'ulot vositalari:** sinf doskasi, plakatlar, fundamental fan darsliklari, o'quv va uslubiy qo'llanmalar, informatika bo'yicha atamalar lug'ati, videoprojektor, ekran va kompyuterdan samarali foydalanish.

**Mashg'ulot usullari:** takrorlash, suhbat va savol-javob, munozara (mavzuni o'zlashtirishni mustahkamlash) tarzida muloqot o'tkazish, (talabalarining mustaqil, erkin fikrlash va so'zlashga o'rgatgan holda fikr mulohazalarini bayon qildirish, buning uchun har bir talabaga, tayanch iboralardan savollar tashlanadi, ular o'z fikrlarini bayon qiladilar, hamma talaba javobni bayon qilib bo'lgandan so'ng talaba bilan birgalikda javoblar yakun qilinadi).

**Darsning xronologik xaritasi** – 80+80 minut.

**Tashkiliy qismi:** Auditoriyaning jixozlanishi va sanitar sharoitlari, talabalar davomati – 2 +2 minut.

**Bilimlarni baholash:** yangi mavzuni o'rganish uchun zarur bo'lgan material bo'yicha suxbat – 10+10 minut.

**Yangi mavzuni bayon etish** – 55+40 minut.

**Mavzu o'zlashtirilgan darajasini aniqlash** – 10+25 minut.

**Uyga vazifa** – 3+3 minut.

**Masala qo'yilishi.** Djek – kompyuterlar sotish bo'yicha agent (kommivoyajer), uning qaramog'ida 20 ta shahar bor. ishlayotgan kompaniya yo'l harajatlarining 50% ni to'laydi. Djek uning qaramog'ida bo'lgan har ikki shahar orasida yo'l harajatini hisoblab chiqqan. Masala yo'l harajatlarini kamaytirishdan iborat.

Dastlabki ma'lumotlar Djek tasarrufidagi shaharlar ruyhati va narxlar matrisasi ko'rinishida berilgan. Bu yerda matrisa  $i$  shahardan  $j$  shaharga borish narxiga teng bo'lgan  $c(i,j)$  elementlardan tashkil topgan ikki o'lchamli massiv. Shaharlar soni 20 ta bo'lsa, matrisa -  $20 \times 20$  bo'ladi.

Biz Djekga yo'l harajatlarini kamaytirishga yordam berishimiz kerak. Djekning marshruti o'zi yashagan shahardan boshlanib, qolgan hamma shaharlarni bir martadan o'tib, yana o'z shahriga qaytib kelishi kerak. Demak, biz tuzayotgan ruyhatda har bir shahar faqat bir marta uchrashi kerak, Lekin Djek yashagan shahar ikki marta uchrab, ruyhatning birinchi va oxirgi elementlari bo'ladi. Undan tashqari, ruyhatdagi shaharlar tartibi Djekning marshrutini belgilaydi. Ruyhatdagi ikkita oxirgi shaharlar orasidagi yo'l narxi – bu butun marshrut narxi deb hisoblanadi. Demak, agar biz Djekga eng kichik narxdagi ruyhatni tuzib bersak, masalani yechgan bo'lamiz.

### **Evristik algoritmlar.**

Evristika yoki evristik algoritm – algoritm deb ta'riflanishi uchun quyidagi hususiyatlarga ega bo'lishi kerak:

3. U odatda shartli ravishda optimal bo'lmasa ham, yahshi yechimlarni topish kerak.

4. Uni ixtiyoriy ma'lum aniq algoritmdan ko'ra, amalga oshirish tezroq va soddaroq bo'lishi kerak.

Odatda yahshi algoritmlar evristik bo'lib chiqadi. Faraz qilaylik, biz tez ishlaydigan va barcha test topshiriqlariga javob beradigan algoritmni tuzdik, lekin uning to'g'riligini isbotlab bilmaymiz. Shunday isbot berilmaguncha, algoritm evristik deb tushuniladi.

Misol tariqasida quyidagi algoritmni ko'rib chiqamiz:

**GTS algoritmi:** (kommivoyajer).  $N$  ta shaharlar va  $C$  narxlar matrisasi berilgan kommivoyajer masalasi uchun  $U$  shahardan boshlab  $COST$  narxli  $TOUR$  yaqinlashgan yechimni topish kerak.

**Qadam 0:** [Insiallashtirish]  $TOUR:=0$ ;  $COST:=0$ ;  $V:=U$ ;

**Qadam 1:** [Hama shaharlarni o'tish] For  $k:=1$  to  $N-1$  do qadam 2;

**Qadam 2:** [Keyingi vektorga o'tish]

Faraz qilaylik,  $(V,W)$  –  $V$  shahardan  $W$  ga olib borayotgan eng kichik narxli vektor. Unda:

$TOUR:=TOUR+(V,W)$ ;  $COST:=COST+C(V,W)$ ;

$V:=W$ ;

**Qadam 3:** [Marshrutni tugatish]  $TOUR:=TOUR+(V,1)$ ;

$COST:=COST+C(V,1)$ ;

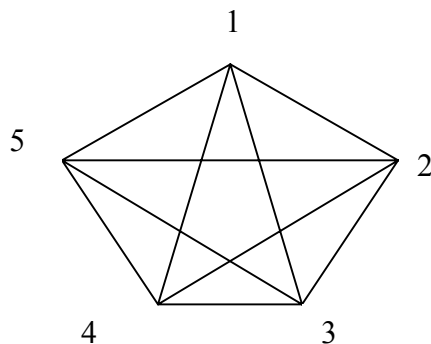
Marshrutni tasvirlash uchun biz matematikada graf yoki tur deb nomlanayotgan chizmadan foydalanamiz. Umuman tur – bu nuqtalar va bir nechta yoki barcha ikki nuqtalarni bog'layotgan chiziqlar to'plami, undan tashqari chiziqlar ustida qiymatlar ham berilishi mumkin.

Masalani soddalashtirish uchun beshta shahar uchun yechim topamiz. Rasm. 1a – narxlar matrisasi. Rasm. 1b – turli model ko'rsatilgan.

-	1	2	7	5
1	-	4	4	3
2	4	-	1	2
7	4	1	-	3
5	3	2	3	-

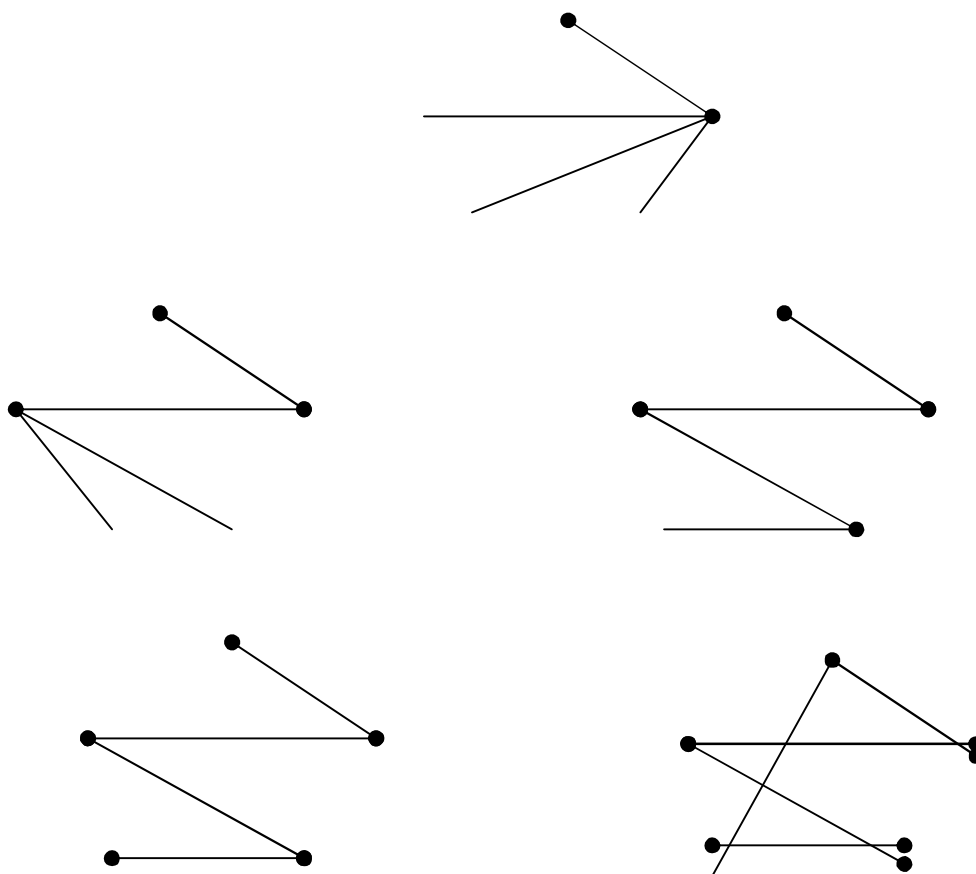
Rasm 1-a). Narxlar matrisasi





Rasm 1-b). To'rsimon model

To'rlar nazariyasida shaharlar ruyhati bir shahardan boshlab va o'sha shaharga barcha qolgan shaharlarni bir martadan o'tib qaytib kelish jarayonini belgilaydi. Bunday o'tishni marshrut deb ta'riflaymiz. Marshrut narxi chiziqlar ustidagi qiymatlar yig'indisi bilan aniqlanadi. Rasm 2 algoritm GTS bo'yicha K marshrutni shahar1 dan boshlab tuzishni aks ettiradi.



Rasm 2. Algoritm qadamlari

GTS algoritmi bo'yicha marshrut narxi 14 ga teng. Bu yerda algoritm eng kichik narxli marshrutni topmaganini ko'ramiz. Masalan, marshrut 1-5-3-4-2-1 narxi  $5+2+1+4+1=13$ .

Odatda yaqinlashgan algoritmlar tez bo'lsa ham, hamma vaqt optimal yechimni berolmaydilar. GTS algoritm uchun biz nazoratchi nisol topib bildik. Lekin, yaqinlashgan algoritm ishlamasligini isbotlash hamma vaqt ham oson bo'lmaydi.

GTS algoritmi uchun dastur yozish ancha yengil. Lekin uni tezligini tahlil qilib ko'raylik. Ixtiyoriy kommivoyajer masalasi uchun (besh shahardan iborat) C narxlar matrisasini o'qish va tuzishga  $O(N^2)$  operatsiya kerak. Demak, pastki murakkablik chegara algoritmi uchun  $O(N^2)$  teng va GTS algoritmini yahshi evristik algoritmi deb qabul qilishimiz mumkin.

### Takrorlash uchun nazorat savollari

1. Masala qo'yilishidagi o'zgaruvchilarni aniqlang.
2. Evristik algoritmlarni ta'riflab bering.
3. GTS algoritmini tuzishdagi qadamlarni aytib bering.
4. Algoritmni baholash jarayonini tavsiflab bering.

### Mustaqil ishlash uchun nazorat savollari:

1. Algoritmni baholash uchun qo'llanishi mumkin bo'lgan mezonlarni tavsiflab bering.
2. Vaqtli mezon bo'yicha baholash jarayoniga misollar ko'rsating.
3. Hajmiy mezon bo'yicha baholash jarayoniga misollar ko'rsating.
4. Evristik usul bilan tuzilgan algoritmlarga 5ta misol ko'rsating.
5. Kommivoyajer masalasining 3ta turli mezon bo'yicha yechim misollarini korsating.

### Mavzuga doir testlar:

1. Quyidagi algoritmda siklning operatorlari necha marta bajariladi?

```
m:=36; n:=56
while m<>n do
if m>n then m:=m-n
else n:=n-m;
```

- A) 6
- B) 4
- C) 1
- D) 8

2. Agar o'zgaruvchilar tavsiflanishi

Type room=1. .30;

Var x: real; y: byte; z: room;

bo'lsa, xatosiz bajarilayotgan buyruşlarni toping.

- A) Z:=30
- B) Z:=x
- C) x:=12; z:=x;
- D) X=y; z:=x

3. X va U uzgaruvchilarning dastlabki qiymatlari mos ravishda 0.9 va -1.5. Quyidagi shartli operator IF X<Y THEN X:=Y ELSE Y:=X; bajarilgandan sung ularning kiymati nimaga teng buladi?

- A) X=0.9 ; Y=0.9
- B) X=0.9 ; Y=-1.5
- C) X=-1.5; Y=0.9
- D) X=-1.5; Y=-1.5

4. Quyidagi

$$N = \begin{cases} 1, \text{ agar } y \geq 0 \text{ va } x \geq 0 \\ 2, \text{ agar } y \geq 0 \text{ va } x < 0 \\ 3, \text{ agar } y < 0 \text{ va } x < 0 \\ 4, \text{ agar } y < 0 \text{ va } x \geq 0 \end{cases}$$

ifoda qiymatini hisoblash uchun keltirilgan shartli operatorlardan kaysi biri tugri?

- A) Keltirilgan operatorlardan hech biri berilgan ifodani tugri hisoblamaydi.  
B) If  $y < 0$  then Begin  $x < 0$  then  $N := 3$  else  $N := 4$  End  
else If  $x < 0$  then  $N := 2$  else  $N := 1$  ;  
C) If  $(y \geq 0)$  and  $(x \geq 0)$  then  $N := 1$  else  $N := 2$  ;  
If  $(y < 0)$  and  $(x < 0)$  then  $N := 3$  else  $N := 4$  ;  
D) If  $x < 0$  then Begin  $y < 0$  then  $N := 1$  else  $N := 2$  End  
else Begin If  $y \geq 0$  then  $N := 3$  else  $N := 4$  End ;
5. Ta'minlash operatori kanday ishni bajarish uchun muljallangan? Eng umumiy javobni toping.

- A) Operatorning ung kismida turgan ifodani hisoblaydi va uning qiymatini chap kismidagi uzgaruvchiga ta'minlaydi.  
V) Uzgaruvchilarga qiymat ta'minlaydi.  
S) Uzgaruvchilarning turini boshkasiga uzgartiradi.  
D) Ifoda qiymati qaysi turga mansubligini aniqlaydi.

#### Adabiyotlar

1. В.А.Успенский, А.Л.Семенов. Теория алгоритмов: основные открытия и приложения. – М: Наука, 1987, 287 с.
2. Т..Кормен, Ч.Лейзерсон, Р.Ривест. Алгоритмы: построение и анализ. Сер: Классические учебники. М.: МЦНМО, 2001.- 960 с.
3. Тыгуу Х. Концептуальное программирование. М: Наука, 1984.
4. Н. Вирт. Алгоритмы и структуры данных. – Досса, Хамарайан, 1997.

## 11 MA'RUZA: ENG QISQA YO'LLARNI TOPISH. DEYKSTRA ALGORITMLARI

### *Reja*

#### **1. Eng qisqa yo'llar masalalarining turlari.**

##### **1. Sozli algoritmni tuzish.**

##### **3. Algoritmni psevdokodda ishlab chiqish**

##### **4. Algoritmni baholash.**

**Darsning maqsadi:** talabalarga algoritmlarni ishlab chiqishning evristik usullari haqida tushuncha berish. Kommivoyajer masalasi yordamida aniq algoritmgga misol ko'rsatish Algoritmni baholash va optimallashtirish bo'yicha kunikma hosil qilish

**Tayanch iboralar:** algoritmlar nazariyasi, evrisika, marshrut, minimum, maksimum, murakkablik, vaqti, hajmiy, mezon, chegara, optimallashtirish, test, hujaatlashtirish.

**Mashg'ulot vositalari:** sinf doskasi, plakatlar, fundamental fan darsliklari, o'quv va uslubiy qo'llanmalar, informatika bo'yicha atamalar lug'ati, videoprojektor, ekran va kompyuterdan samarali foydalanish.

**Mashg'ulot usullari:** takrorlash, suhbat va savol-javob, munozara (mavzuni o'zlashtirishni mustahkamlash) tarzida muloqot o'tkazish, (talabalarning mustaqil, erkin fikrlash va so'zlashga o'rgatgan holda fikr mulohazalarini bayon qildirish, buning uchun har bir talabaga, tayanch iboralardan savollar tashlanadi, ular o'z fikrlarini bayon qiladilar, hamma talaba javobni bayon qilib bo'lgandan so'ng talaba bilan birgalikda javoblar yakun qilinadi).

**Darsning xronologik xaritasi** – 80 minut.

**Tashkiliy qismi:** Auditoriyaning jixozlanishi va sanitar sharoitlari, talabalar davomati – 2 minut.

**Bilimlarni baholash:** yangi mavzuni o'rganish uchun zarur bo'lgan material bo'yicha suxbat – 10 minut.

**Yangi mavzuni bayon etish** – 55 minut.

**Mavzu o'zlashtirilgan darajasini aniqlash** – 10 minut.

**Uyga vazifa** – 3 minut.

Yo'l tarmoqlari atlas (karta) qismi berilgan bo'lib, undan A va B nuqtalar orasidagi "eng yahshi" marshrutni topish kerak bo'lsin. "Eng yahshi" marshrutni ko'p faktorlar belgilashi mumkin, masalan, tezlik cheklangan holda marshrutni o'tish vaqti, o'tish kerak bo'lgan shaharlar soni va boshqalar.

Biz masalani eng qisqa yo'llar faktori bo'yicha yechamiz. Masalaning modeli turlar yordamida tuziladi. Uzluksiz G turni har bir qirrasiga uning uzunligiga teng qiymat berilgan ko'rinishida tuzamiz. Bunday turda masofa irrallar yig'indisiga teng bo'ladi. Masalaning maqsadi ikkita berilgan uchlar orasidagi eng qisqa marshrutni topishdir.

Umuman, eng qisqa yo'llar masalalari kombinator optimallashtirishning fundamental muammolaridandir. Ularning bir necha turlari mavjud, masalan, ikkita berilgan uchlar orasida, berilgan va qolgan barcha uchlar orasida, turdagi har bir uchlar juftliklari orasida va boshqalar.

### **Deykstra algoritmning so'zli tavsifi**

Shunday masalalarni yechish uchun Deykstra algoritmi ancha qulay va yahshi deb topilgan. Algoritm quyidagi qadamlardan iborat:

6. Dastlab, berilgan **(Lex)** uchidan qolgan barcha uchlargacha bir qirra uzunligidagi masofalar aniqlanadi.
7. Ulardan eng qisqasi "doimiy eng qisqa masofa" sifatida belgilanadi (**Lex** va **BVa** uchlari qirras).
8. Aniqlangan masofa **BVa** dan boshqa bor uchlargacha masofalarga qo'shiladi.

9. Hosil bo'lgan yig'indilar dastlab aniqlangan **Lex** dan qolgan uchlargacha bo'lgan masofalar bilan taqqoslanadi. Natijada masofasi qisqaroq bo'lgan uchning qirradi tanlanadi.
10. **BVa** uchi, eng qisqa masofa aniqlangan uch sifatida, ruyhatdan o'chiriladi. Ruyhatga boshqa uch qo'yiladi, masalan, **Roa. Bva** o'z navbatida, boshqa, izlanayotgan ruyhatga qo'yiladi.

Keyingi eng qisqa masofani topish uchun butun jarayon qayta bajariladi. **BVa** dan keyin yana bir uch ruyhatga qo'yiladi. Dastlabkisi esa ruyhatdan o'chiriladi. Sikl **Bed** va **Lex** uchlarini bog'lash uchun belgilangan qirralar aniqlanishi bilan to'xtatiladi.

Rasm bo'yicha ikkinchi iteratsiyada **Nbr** uchi aniqlanadi va **Roa** gacha masofa 41 deb qabul qilinadi. Uchinchi iteratsiyada **Gla** uchigacha masofa eng qisqa va 27 deb qabul qilinadi. Quyidagi rasmda eng qisqa yo'llar daraxti ko'rinishida ularning natijaviy to'plami keltirilgan.

Aylanalar ichidagi sonlar algoritm bo'yicha qirralar tanlanish tartibini ko'rsatadilar. Bu daraxt bo'yicha biz **Lex** uchidan ixtiyoriy bizni qiziqtirayotgan uchgacha eng qisqa yo'lni topishimiz mumkin.

Ko'rilgan misolda **Bed** uchi **Lex** dan boshlab eng oxirgi bo'lib chiqdi, ya'ni **Lex** dan **Bed** gacha eng qisqa masofani topish uchun biz **Lex** dan barcha qolgan uchlargacha eng qisqa yo'llarni topishga majbur bo'ldik.

Demak, eng yomon holatda 2 ta berilgan uchlar orasidagi eng qisqa yo'lni topish, bir berilgan nuqtadan qolgan barcha nuqtalargacha eng qisqa yo'l topish masalasi bilan murakkabligi bir xil bo'ladi.

### Algoritmni psevdokodda ishlab chiqish

#### 2. Masala qo'yilishi.

$M$  ta uch va  $N$  ta qirralardan iborat uzluksiz grafda  $V_0$  uchidan  $W$  uchigacha  $Dist(W)$  masofani topish kerak. Qirralar uzunliklari  $A$  matrisa bilan berilgan deb hisoblaymiz.

**Qadam 0.** [uchlarni belgilash] – bu yerda  $V_0$  uchini “aniqlangan” deb belgilaymiz, qolgan barcha uchlarini “aniqlanmagan” deb hisoblaymiz.

**Qadam 1.** [o'zgaruvchilarni inetsiallashtirish] – bu yerda  
 $Dist(U) := A(V_0, U)$  – barcha  $G$  ga tegishli  $U$  uchlar uchun;  
 $Dist(V_0) := 0$ ;  $Next := V_0$ ;

**Qadam 2.** [sikl]. *While*  $Next \neq W$  *do*

*Begin*

**Qadam 3.** [“aniqlanmagan” uchgacha eng qisqa yo'lni yangilash]. Har bir “aniqlanmagan”  $U$  uchi uchun

$Dist(U) := \min(Dist(U), Dist(Next) + A(Next, U))$ .

**Qadam 4.** [“aniqlanmagan” uchgacha eng qisqa yo'lni tanlash]. Agar  $U$  barcha “aniqlanmagan” uchlar orasida  $Dist(U)$  masofasi eng kichik bo'lsa, uni “aniqlangan” deb belgilaymiz va  $Next := U$ .

**end.**

Bu algoritmning va dasturning murakkabligini  $O(M^2)$  ekanligini ko'rsatish mumkin.

### Takrorlash ucun nazorat savollari

1. Qaysi mezonlar bo'yicha eng qisqa yo'llar masalalarini yechish mumkin?
2. Deykstra algoritmi nima uchun yaxshi hisoblanadi?
3. Algoritmni psevdokodda tavsiflashning qo'layligini ko'rsating
4. Deykstra algoritmining bahosi qanday?

### Mustaqil ishlash uchun nazorat savollari:

1. Algoritmni baholash uchun qo'llanishi mumkin bo'lgan mezonlarni tavsiflab bering.
2. Vaqtli mezon bo'yicha baholash jarayoniga misollar ko'rsating.
3. Hajmiy mezon bo'yicha baholash jarayoniga misollar ko'rsating.
4. Eng qisqa yo'llarni topish masalasiga 3ta turli mezon bo'yicha yechim misollarini korsating.
5. Deykstra algoritmidan farqli boshqa eng qisqa yo'llarni topadigan algoritmni tuzing.

### Mavzuga doir testlar:

1. Quyida ikki algoritm keltirilgan:

1-algoritm: boshlanish  $i:=100$ ,  $S1:=1$ ; toki  $i \geq 1$  takrorlash boshlanish  $S1:=S1+i$ ;  $i:=i-1$  tamom; chikarish  $S1$ ; tamom.

2-algoritm: boshlanish  $i:=100$ ,  $S2:=1$ ; toki  $i \geq 1$  takrorlash boshlanish  $S2:=S2*i$ ;  $i:=i-1$  tamom; chikarish  $S2$ ; tamom.

Birinchi va ikkinchi algoritm bajarilishi natijasida mos ravishda  $S1$  va  $S2$  kiymatlar xosil kilinadi.  $S1$  va  $S2$  urtasida kuyidagi keltirilgan munosabatlardan kaysi biri bajariladi?

- A)  $S1 < S2$ ;
- B)  $S1 > S2$ ;
- C)  $S1 = S2$ ;
- D)  $S1 = 2 * S2$ ;

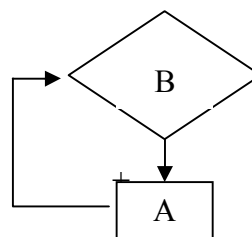
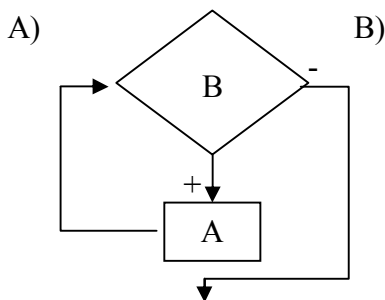
2. Obyektga yunaltirilgan dasturlashning asosiy goyasi?

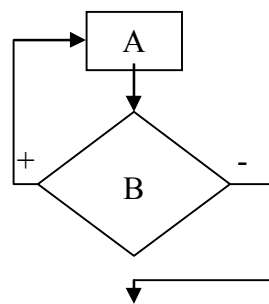
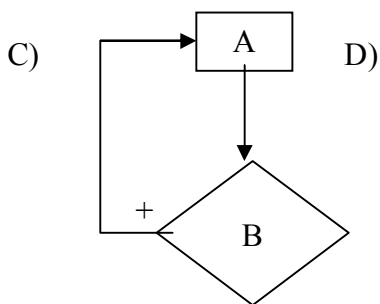
- A) ma'lumotlar va ular ustida bajariladigan amallarni bir strukturaga birlashtirish;
- V) ma'lumotlarni obyektlar sifatida tavsiflash;
- S) ma'lumotlar va ular ustida bajariladigan amallarni aloxida-aloxida dasturlash;
- D) obyektlar turi degan tushunchani kiritish

18. Obyektga yunaltirilgan dasturlash kuyidagi uch tushunchaga asoslanadi:

- A) inkapsulyasiya; merosxurlik

3. Old-shartli sikl operatori While B do A (bu yerda V-mantikiy turdagi ifoda, A-oddiy yoki murakkab operator)ning bajarilish jarayonini ifodalovchi blok-sxemani kursating.



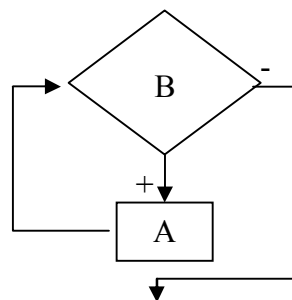
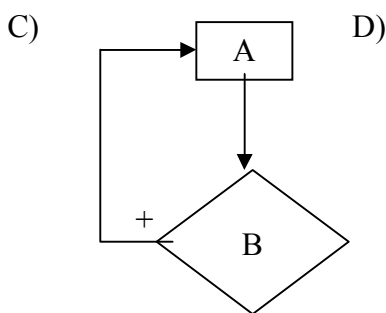
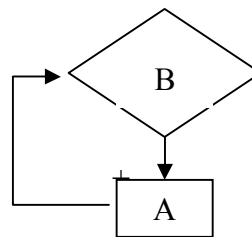
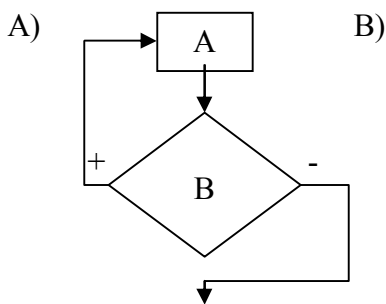


4. Quyidagi operatorlarning bajarilishidan sung uzgaruvchining qiymati nimaga teng buladi?

- 1) I:=1; F:=2; while I<6 do I:=I+1; F:=F\*I;  
 2) I:=1; F:=2; while I<6 do begin I:=I+1; F:=F\*I End;

- A) 1) F=12;                      2) F=1440;  
 B) 1) F=48 ;                    2) F=240;  
 C) 1) F=240;                    2) F=48;  
 D) 1) F=14;                      2) F=11080.

5. Sung shartli sikl operatori Repeat A Until B ga mos keluvchi blok-sxemani kursating (bu yerda A-oddiy yoki murakkab operator, V-mantikiy ifoda).



6. Quyidagi jumalardan kaysi old va sung shartli sikl operatorlari orasidagi farqlardan birini tugri va tulik ifodalaydi?

A) Bu javoblarning birontasi xam old va sung shartli sikl operatorlar urtasidagi farqlardan xech birini tugri ifodalamaydi.

V) Old shartli sikl operatorida sikl jismining takror bajarilishi mantikiy ifoda qiymati "false" bulganda, sung shartli sikl operatorida esa "true" bulganda ruy beradi;

C) Agar sikl jisimida mantikiy ifoda qiymatiga ta'sir kiluvchi operator bulmasa, u xolda sung shartli sikl operatori cheksiz takrorlanishga («zasiklivaniye») olib kelish mumkin. Old shartli sikl operatorida esa bu xodisa xech kachon ruy bermaydi;

D) Old shartli sikl operatorida mantikiy ifoda qiymati birinchi xisoblashdayok "false" ga teng buladi, u xolda sikl jisimi bir marta xam bajarilmaydi. Sung shartli sikl operatorida esa sikl jisimi mantikiy qiymatdan boglik bulmagan xolda xech bulmasa bir marta albatta bajariladi;

## Adabiyotlar

1. В.А.Успенский, А.Л.Семенов. Теория алгоритмов: основные открытия и приложения. – М: Наука, 1987, 287 с.
2. Т.Кормен, Ч.Лейзерсон, Р.Ривест. Алгоритмы: построение и анализ. Сер: Классические учебники. М.: МЦНМО, 2001.- 960 с.
3. Гуломов С.С. ва бошқалар. Ахборот тизимлари ва технологиялари. Тошкент, 2000 й.
4. Жуманов И.И. Мингбаев Н.С., Информатика.- Самарқанд: СамДУ нашри, 2002, 107 бет.
5. Ahatov A.R., Zaripova G.L. va boshq. Axborot texnologiyalari // Uslubiy qo'llanma. – Samarqand: SamDU nashri, 2008 yil – 112 bet.
6. Интеллектуализация ЭВМ. Перспективы развития вычислительной техники. Под ред. Ю.М.Смирнова. М: 1989 г.
7. Тыугу Х. Концептуальное программирование. М: Наука, 1984.
8. Н. Вирт. Алгоритмы и структуры данных. – Досса, Хамарайан, 1997.
9. Жуманов И.И., Мингбоев Н.С. Ҳисоблаш системаларининг информацион асослари. Самарқанд: СамДУ нашри, 2002, 107 бет.

## 12 MA'RUZA: TARTIBLASH ALGORITMLARI. TEZ TARTIBLASH

### *Reja*

1. *Tartiblash masalalarining turlari*
2. *Xoaraning tartiblash algoritmi mazmuni*
3. *Xoara algoritmini rekursiv usulda amalgam oshirish*
4. *Algoritmni baholash*

**Darsning maqsadi:** talabalarga tartiblash masalalarini yechaydigan algoritmlar haqida tushuncha berish. Xoaraning tartiblash algoritmi mazmunini yetkazish, algoritmni baholash va optimallashtirish bo'yicha kunlikma hosil qilish

**Tayanch iboralar:** algoritmlar nazariyasi, tartiblash, minimum, maksimum, murakkablik, massiv, o'lcham, chegara, optimallashtirish, test, hujaatlashtirish.

**Mashg'ulot vositalari:** sinf doskasi, plakatlar, fundamental fan darsliklari, o'quv va uslubiy qo'llanmalar, informatika bo'yicha atamalar lug'ati, videoprojektor, ekran va kompyuterdan samarali foydalanish.

**Mashg'ulot usullari:** takrorlash, suhbat va savol-javob, munozara (mavzuni o'zlashtirishni mustahkamlash) tarzida muloqot o'tkazish, (talabalarining mustaqil, erkin fikrlash va so'zlashga o'rgatgan holda fikr mulohazalarini bayon qildirish, buning uchun har bir talabaga, tayanch iboralardan savollar tashlanadi, ular o'z fikrlarini bayon qiladilar, hamma talaba javobni bayon qilib bo'lgandan so'ng talaba bilan birgalikda javoblar yakun qilinadi).

**Darsning xronologik xaritasi** – 80 minut.

**Tashkiliy qismi:** Auditoriyaning jixozlanishi va sanitar sharoitlari, talabalar davomati – 2 minut.

**Bilimlarni baholash:** yangi mavzuni o'rganish uchun zarur bo'lgan material bo'yicha suxbat – 10 minut.

**Yangi mavzuni bayon etish** – 55 minut.

**Mavzu o'zlashtirilgan darajasini aniqlash** – 10 minut.

**Uyga vazifa** – 3 minut.

### **Tartiblash masalalarining turlari**

Umuman tartiblash deganda berilgan ob'yektlar to'plamini ma'lum tartibda joylash uchun qayta ishlash jarayoni tushuniladi.



Tartiblash natijasida to'plamdagi elementlarni izlash jarayonlari yengillashadi. Undan tashqari tartiblashlar misolida qanday qilib algoritmni murakkablash evaziga samaradorlikni oshirishga erishish mumkinligini ko'rsatsa bo'ladi.

Hozirgi kunda ko'pgina tartiblash algoritmlari mavjud. Algoritmni tanlash qayta ishlanayotgan ma'lumotlar strukturasi bog'liq va shu sababli tartiblash usullari asosan 2 sinfga ajratiladi. Bular massivlarni tartiblash va fayllarni tartiblash. Ularni yana ichki va tashqi tartiblash ham deb nomlaydilar. Chunki massivlar mashinaning tezkor xotirasida joylashadi. Fayllar esa odatda ancha hajmi katta bo'lgan lekin sekin ishlaydigan tashqi xotiradan olinadilar.

### Xoaraning tartiblash algoritmi mazmuni

Eng yahshi tartiblash algoritmlaridan biri deb Ch. Xoara usuli hisoblanadi. Bu usul almashuvga asoslangan.

Bu yerda yahshi samaradorlikka erishish uchun dastlab katta masofadagi ya'ni bir-biriga eng uzoq joylashgan elementlarni almashtirish qo'llaniladi. Faraz qilaylik bizda  $n$  ta element kalitlar bo'yicha qayta tartibda berilgan. Xoara usuli bo'yicha ularni  $\frac{n}{2}$  ta almashuv bilan tartiblash mumkin. Buning uchun dastlab eng chap va eng o'ng tomonda joylashgan elementlarni almashtiramiz. Keyin ikki tomondan o'rtaga qarab kelamiz. Lekin bu faqatgina qayta tartib aniq bo'lganda amalga oshiriladi.

Endi massiv ixtiyoriy tartibda berilgan bo'lsin. Ixtiyoriy  $X$  elementni tanlab massivni chapdan o'ngga qandaydir  $a_i > x$  element uchramaguncha ko'rib chiqamiz. Keyin massivni o'ngdan chapga qandaydir  $a_j < x$  element uchramaguncha o'tamiz.

$a_i$  va  $a_j$  elementlarni o'rinlarini almashtirib massivni ikki tomondan ko'rib chiqish jarayonini massiv o'rtasiga kelmaguncha davom ettiramiz. Natijada massiv 2 qismga bo'linadi. Chap qismdagi elementlar  $x$  dan katta yoki teng bo'ladilar. O'ng tomondagi elementlar  $x$  dan kichik yoki teng bo'ladi.

Dastur tuzayotganda bu jarayonni prosedura yordamida amalga oshirish mumkin. Prosedurani rekursiv va norekursiv usullar bilan tuzish mumkin.

### Xoara algoritmini rekursiv usulda amalga oshirish

Quyidagi dastur rekursiv prosedurani qo'llaydi.

Prosedure Hoare;

Prosedure sort (L, R: index);

var i, j: index; w, x: item;

begin i:=L; j:=R; x:=a[(L+R) div 2];

repeat

while a[i]<x do i:=i+1 end;

while a[j]>x do j:=j-1 end;

if i<=j then

begin w:=a[i]; a[i]:=a[j]; a[j]:=w;

i:=i+1; j:=j-1; end;

until i>j

if L<j then sort (L, j) end;

if i<R then sort (i, R) end;

end {\*sort\*};

begin sort (1, n);

end {\* Hoare\*};

Norekursiv dasturni tuzish uchun yordamchi steklardan foydalaniladi.

## Algoritmni baholash

Xoara algoritmni unumdorligini tahlil qilamiz. Boshlab bo'linish jarayonini ko'raylik. Qandaydir  $x$  ni tanlab biz massivni to'liq o'tamiz. Demak,  $n$  ta taqqoslashni amalga oshiramiz. Taqqoslashlarni umumiy soni  $n \cdot \log(n)$  ekanligi, o'rin almashtirishlar soni esa  $\frac{n \cdot \log(n)}{6}$  ekanligi isbotlangan.

Bizning misolimizda  $x$  - o'rtacha element deb tanlangan, lekin Xoara fikri bo'yicha  $X$  ixtiyoriy tanlanishi kerak. Xoara algoritmining o'rtacha ishlash vaqti  $O(n \cdot \ln(n))$  teng.

### Takrorlash ucun nazorat savollari

1. Tartiblash masalalarining qaysi turlarini bilasiz?
2. Xoaraning tartiblash algoritmi mazmuni nimada?
3. Xoara algoritmini qanday usullar bilan amalga oshirish mumkin?
4. Xoara algoritmi bahosini tavsiflab bering.

### Mustaqil ishlash uchun nazorat savollari:

1. Algoritmni baholash uchun qo'llanishi mumkin bo'lgan mezonlarni tavsiflab bering.
2. Turli mezonlar bo'yicha baholash jarayoniga misollar ko'rsating.
3. Tartiblash usullariga 5 ta misol ko'rsating.
4. Xoara tartiblash algoritmi yordamida yechiladigan masalalar turkumini tasniflab bering.
5. Xoara tartiblash algoritmini rekursiv bo'lmagan dasturini tuzing.
6. Xoara tartiblash algoritmini ob'yektli yo'naltirilgan dasturlash tilida amalgam oshiring.

### Mavzuga doir testlar:

1. Quyida parametrlari sikl operatoriga doir misollar berilgan:

Var S1, S2, I:integer;

L:boolean;

1) S1:=1; For I:=3 to 5 do S1:=S1\*I

2) S1:=0; For L:=false to true do S2:=S2+1;

bu operatorlarning bajarilishidan sung S1 va S2 uzgaruvchilarning qiymati nimaga teng buladi?

- |              |           |
|--------------|-----------|
| A) 1) S1=60; | 2) S2=2;  |
| B) 1) S1=12; | 2) S2=3;  |
| C) 1) S1=54; | 2) S2=60; |
| D) 1) S1=30; | 2) S2=20; |

2. Quyidagi jummalardan kaysi biri yolg'on?

- A) MS massivda 7ta element bor; elementlarga murojaat MC [false, 2] kabi buladi;
- B) MV massiv 9 elementdan iborat bulib, uning elementlariga murojaat MV [3] kabi buladi.
- C) ZAR massiv 8 elementdan iborat va unga murojaat ZAR [m1, m3] kabi buladi;
- D) MN massivda 10 element bor: elementlarga murojaat MN[x1] kabi bo'ladi;

3. Tuplamli turning asosiy turi sifatida kaysi turlar ishlatilishi mumkin?

- A) chegaralanmagan butun va xakikiy turdan boshka ixtiyoriy chegaralangan yoki skalyar turlar;
- B) fakat standart turlar;
- C) ixtiyoriy chegaralangan yoki skalyar turlar;
- D) fakat skalyar turlar;

4. Quyidagi bandlardan kaysi birida xato mavjud?

- A) massiv indeks turi sifatida ixtiyoriy standart, skalyar sanokli va chegaralangan turlarni ishlatish mumkin, masalan, integer, boolean, real, char va x.k.
- B) massivni tasvirlash uchun uning elementlari turini va indekslari turini kursatish kerak.
- C) regulyar tur yoki massiv deganda bir xil turdagi ma'lumotlarning tartiblangan tuplamini tushunamiz.
- D) massiv elementlari turi sifatida ixtiyoriy oddiy yoki murakab tur ishlatilishi mumkin.

5. Prosedura sarlavxasi: Procedure <ism> (<formal parametrlar ruyxati>);. <formal parametrlar ruyxati> nimadan iborat bulishi mumkin? Eng umumiy javobni toping.

- A) Parametr-kiymatlar, parametr uzgaruvchilar (ularning oldida Var suzi bulishi kerak), parametr proseduralar (ular oldida Procedure suzi bulishi kerak) va parametr-funksiyalar (ular oldida Function suzi kelishi kerak) va ularning turlari kursatilgan ruyxatidan iborat bulishi mumkin.
- B) uzgarmaslar va uzgaruvchilarning nomlari va turlari kursatilgan ruyxatdan.
- C) Proseduradan ishlatiladigan prosedura va funksiyalarning nomlaridan.
- D) Prosedura ichkarisida ishlatiladigan barcha uzgaruvchilarning turlari kursatilgan ruyxatidan.

6. SS1 ismli prosedura sarlavxasi Procedure SS1 (a,b,c: real; i: integer; d: char; k: boolean; Var t: char); va asosiy dasturning uzgaruvchilar bulimi

Var: x, y, z: real; k, l: integer; w1, w2: char; n3: boolean;  
berilgan. Kuyidagi SS1 prosedura murojaat kiluvchi operatorlar keltirilgan. Ulardan kaysi biri tugri?

- A) SS1 (x, y, z, l, w1, n3, w2);
- B) SS1 (x, n3, z, l, w1, y, w2);
- C) SS1 (x, y, z, l, w2, n3);
- D) SS1 (x, y, z, l:integer, d, t: char, n3);

7. Prosedura sarlavxasida nimalar kursatiladi?

- A) Prosedura ismi va formal parametrlar ruyxati;
- B) Prosedura operatori;
- C) prosedura ismi;
- D) Anik parametrlar

8. Proseduraning formal parametrlar ruyxati nimalardan iborat bulishi mumkin?

- A) Formal parametrlar ismlari va turlari kursatilgan ruyxatdan;
- B) Prosedura operatoridan;
- C) Anik parametrlar ismlari;
- D) Prosedura ismidan;

9. Prosedura parametrlari kanday maksadda ishlatiladi?

- A) Proseduraga boshlangich ma'lumotlarni va proseduradan uni chakiruvchi dasturga natijalarni uzatish uchun.
- B) Prosedurani tavsiflash uchun;
- C) Uzgaruvchilarni tavsiflash uchun;
- D) Nishonlarni uzatish uchun;

#### **Adabiyotlar**

1. В.А.Успенский, А.Л.Семенов. Теория алгоритмов: основные открытия и приложения. – М: Наука, 1987, 287 с.
2. Т.Кормен, Ч.Лейзерсон, Р.Ривест. Алгоритмы: построение и анализ. Сер: Классические учебники. М.: МЦНМО, 2001.- 960 с.
3. Гуломов С.С. ва бошқалар. Ахборот тизимлари ва технологиялари. Тошкент, 2000 й.
4. Жуманов И.И. Мингбаев Н.С., Информатика.- Самарқанд: СамДУ нашри, 2002, 107 бет.
5. Ahatov A.R., Zaripova G.L. va boshq. Axborot texnologiyalari // Uslubiy qo'llanma. – Samarqand: SamDU nashri, 2008 yil – 112 bet.
6. Интеллектуализация ЭВМ. Перспективы развития вычислительной техники. Под ред. Ю.М.Смирнова. М: 1989 г.
7. Тыгу Х. Концептуальное программирование. М: Наука, 1984.
8. Н. Вирт. Алгоритмы и структуры данных. – Досса, Хамарайан, 1997.
9. Жуманов И.И., Мингбоев Н.С. Ҳисоблаш системаларининг информацион асослари. Самарқанд: СамДУ нашри, 2002, 107 бет.

### **13 MA'RUZA: JADVAL – AXBOROT STRUKTURASI ELEMENTI SIFATIDA**

#### **R E J A:**

1. Jadval ko'rinishidagi axborot srykturasini boshqarish tizimlari.
2. Ma'lumotlarning turlari va tuzilishi.
3. Jadvalda ma'lumotlarni kodlash. Jadvalning bosh kaliti.
4. Ma'lumotlar omboriga pelyatsion yondashish muammolari.

**Darsning maqsadi:** Talabalarga ma'lumotlar ombori, ma'lumotlarni boshqarish, ob'ekt va kattalik, kodlash tushunchasi, ma'lumotlarning turlari va ularning tuzilishi, ma'lumotlarni boshqarish tizimlari haqida ma'lumot berish.

**Tayanch iboralar:** Ma'lumotlar ombori, ob'ekt, kattalik, ma'lumotlar omborini boshqarish tizimlari, sonli ma'lumotlar, belgili ma'lumotlar, tarmoqli tizim, jadvalli, mantiqiy ma'lumotlar, ierarxik tizim, tuzilish, mantiqiy model, pog'onali, tarmoqli, relyatsion tizim, normallashtirish.

**Mashg'ulot vositalari:** sinf doskasi, plakatlar, fundamental fan darsliklari, o'quv va uslubiy qo'llanmalar, informatika bo'yicha atamalar lug'ati, videoprojektor, ekran va kompyuterdan samarali foydalanish.

**Mashg'ulot usullari:** takrorlash, suhbat va savol-javob, munozara (mavzuni o'zlashtirishni mustahkamlash) tarzida muloqot o'tkazish, (talabalarning mustaqil, erkin fikrlash va so'zlashga o'rgatgan holda fikr mulohazalarini bayon qildirish, buning uchun har bir talabaga, tayanch iboralardan savollar tashlanadi, ular o'z fikrlarini bayon qiladilar, hamma talaba javobni bayon qilib bo'lgandan so'ng talaba bilan birgalikda javoblar yakun qilinadi).

**Darsning xronologik xaritasi** – 80 minut.

**Tashkiliy qismi:** Auditoriyaning jixozlanishi va sanitar sharoitlari, talabalar davomati – 2 minut.

**Bilimlarni baholash:** yangi mavzuni o'rganish uchun zarur bo'lgan material bo'yicha suxbat – 10 minut.

**Yangi mavzuni bayon etish** – 55 minut.

**Mavzu o'zlashtirilgan darajasini aniqlash** – 10 minut.

**Uyga vazifa** – 3 minut.

### **Ma'lumotlarni boshqarish tizimlari**

Shaxsiy kompyuterlar hisoblash texnikasi vositalarining yangi sinfini tashkil etadi. Ular axborotlarni qayta ishlashning yuqori kafolatlanishi, arzonligi, ixchamligi, turli amallarni bajara olishi, kam elektr quvvatini sarflashi bilan ajralib turadi. Bu xususiyatlar kompyuterlar asosida keng vazifalarga mo'ljallangan avtomatlashgan ishchi joylari (AIJ)ni yaratish imkonini beradi.

Kompyuterda ma'lumotlar omborini hosil qilish va ular bilan ishlash uchun maxsus dasturlar yaratiladi. Keyingi yillarda chiqarilayotgan kompyuterlarda amal bajarish tezligining juda kattaligi (100 mlrd/s) va uning xotirasi kengayganligi (40-60 Gb) katta imkoniyatlarga ega bo'lgan ma'lumotlar omborini yaratish imkonini bermoqda.

Kompyuterda ma'lumotlar omborini yaratish va uni ishlatishni belgilovchi ikkita omil mavjud: birinchi omil - qanday mazmunga ega bo'lgan ma'lumotlar (ya'ni ularni tashkillashtirish) va ikkinchisi - yig'ilgan ma'lumotlardan qanday foydalanish kerakligidir. Masalan, xodimlar ro'yxatini qayta ishlavchi turli xildagi dasturlar mavjud. Lekin bunday dasturlar biror kishining familiyasini izlash yoki u haqda biror ma'lumotni topishga (ajratishga) qarab bir-biridan jiddiy farq qiladi. Bundan tashqari, har bir berilgan soha uchun o'zining dasturi ishlab chiqilishi kerak. Masalan, kimyoviy moddalar haqidagi ma'lumotlar ombori uchun mo'ljallangan dasturlar aviapassajirlar ro'yxatini qayta ishlash uchun umuman yaroqsizdir.

Ma'lumotlar omboridan foydalanish uchun maxsus dasturlar yaratiladi va bunday dasturlar **ma'lumotlar omborini boshqarish tizimi** deb ataladi.

Ma'lumotlar omborida axborotlar asosan matn va raqam ko'rinishida saqlanadi.

Ma'lumotlarni boshqarish tizimi vazifasiga ma'lumotlar omborini boshqarishning quyidagi xususiyatlari kirishi mumkin:

\* Ma'lumotlar omboriga kirish: foydalanuvchining talabiga javoban axborot turidan qat'i nazar unga qulay ko'rinishda javob berish.

\* Ma'lumotlarni modifikatsiyalash: berilgan axborotni foydalanuvchi talabiga mos holda (texnik imkoniyatlari hisobga olingan holda) o'zgartirish.

\* Ishonchlilik darajasi: qurilmalar tasodifan to'xtatilganda ma'lumotlar omborining qayta tiklanish qobiliyati.

\* Ma'lumotlarni himoyalash: ma'lumotlar omboridan ruxsatsiz (sanktsiyasiz) foydalanishning cheklanganligi.

\* Ma'lumotlar omboridan tarmoqda foydalanish: ma'lumotlardan bir vaqtda bir necha kishining (bir-biriga xalaqit bermasdan) foydalanishi.

Ma'lumotlar omborini boshqarish tizimi o'zining ma'lum ko'rinishdagi ichki tuzilishiga, amalga oshiruvchi amaliy dasturlarga ega bo'lishi kerak.

Bunday dasturlar ikki turga: integrallashgan (umumlashtirilgan) va paketli dasturlarga ajratiladi.

Integrallashgan dasturlar avtomatlashgan ishchi joylari yaratishning asosini tashkil etadi. Bunday tizimlar quyidagilarni amalga oshirishni ta'minlaydi:

\* muammoli - yunaltirilgan axborotlarni kiritishning qulayligi;

\* oldin kiritilgan axborotlardan foydalanishning engilligi; murakkab strukturali hujjatlarni qayta ishlash va shakllantirish;

\*shaxsiy kartotekalar, ishchi kalendarlar, yozuv daftarlari va boshqa vositalarni yaratishning mumkinligi.

Hozirgi kunda integrallashgan dasturlarga nisbatan paketli dasturlardan ko'proq foydalaniladi.

Keng tarqalgan paketli dasturlarga matnli axborotlarni qayta ishlash (Word), «Elektron jadvallar» va ma'lumotlar omborini qayta ishlash tizimlari misol bo'ladi. Keng tarqalganligi bo'yicha «Ishchi grafiklar» tizimi va kommunikatsiya tizimlari keyingi o'rinda turadi.

Muayyan yo'nalishdagi ma'lumotlar omborini boshqarish uchun mutaxassislar tomonidan dasturlar yaratiladi. Ularda foydalanuvchining tizim bilan muloqoti asosida ma'lum hujjatlarni yaratish yoki uni qayta ishlashda ko'riladigan tipik vazifalar (imkoniyatlar majmui) yoritib beriladi.

### **Ma'lumotlarning turlari va tuzilishi**

Kompyuterda qayta ishlanadigan ma'lumotlar o'z ichiga kundalik turmushda ishlatiladigan barcha axborotlarni oladi. Albatta, turli kasb egalari o'zlariga kerakli bo'lgan ma'lumotlar bilan ish yuritadi. Bunday ma'lumotlarning eng asosiylari sonli (raqamli) va belgili (matnli) axborotlar hisoblanadi. Odatda, har qanday axborot tizimini yaratish uchun bu ikki turdagi ma'lumot shakli etarlidir, chunki foydalanuvchiga etkazilmoqchi bo'lgan barcha axborot raqamlar yoki so'zlardan iborat bo'ladi. Shunday bo'lsada, axborotlar tizimini yaratuvchilar yana ikkita ma'lumot turi - **sanalar** va **mantiqiy** o'zgaruvchilardan xabardor bo'lishi lozim.

Ma'lumotlarning turlarini ko'rib chiqamiz.

**Belgili ma'lumotlar.** Har qanday belgili ma'lumotning qiymati ixtiyoriy uzunlikdagi alifbo - raqamli belgilar majmuidan iborat (masalan, «5», «Vatan», «10+25», ...). Ma'lumotlarni bunday ifodalash eng ko'p tarqalgan. Belgili ma'lumotning qiymati «1055» ga teng bo'lsa, mashina bu qiymatni osongina songa aylantirib olishi ham mumkin.

Ko'pgina tizimlarda belgili ma'lumotning uzunligi 255 belgi bilan chegaralangan. Lekin ishlab chiqarish texnologiyasining rivojlanishi axborotlar tizimi yaratuvchilariga keng imkoniyatlar ochib bermoqda. Hozirgi paytda ikki o'lchamli jadvallardan, ba'zi tizimlarda belgili ma'lumot qiymati sifatida uzunligi 30 yoki 40 belgi bo'lgan biror turdagi fayl nomini yozish imkoniyatlari mavjud.

Bu nom orqali ixtiyoriy uzunlikdagi ob'ektning biografiyasini o'z ichiga olgan matnli fayl yoki ob'ektning fotosuratini ko'rsatuvchi grafik fayl ochish mumkin.

Bundan tashqari, hozirgi zamon tizimlarida belgili ma'lumot sifatida ixtiyoriy tuzilish va uzunlikdagi (shu jumladan, tasvir va ovoz) ob'ektlar saqlanishi mumkin.

**Sonli ma'lumotlar.** Bunday turdagi ma'lumotlarga faqat sonlar misol bo'ladi. Axborotlar tizimi foydalanuvchilari uchun bunday ma'lumotlar faqat o'lchami (raqamlar soni) va uning aniqligi (nuqtadan keyingi raqamlar soni)ning ahamiyati katta.

**Mantiqiy ma'lumotlar.** Bu turdagi ma'lumotlar (ba'zida bunday ma'lumotlar Bul qiymatlari deyiladi) bir-birini inkor qiluvchi TRUE (rost) «1» yoki FALSE (yolg'on) «0» qiymatlarini qabul qiladi.

Ulardan tizimda ko'zda tutilgan maqsadni amalga oshirish uchun biror ob'ektning holatini ko'rsatishda foydalanish mumkin. Masalan, «lampa yoniq» (TRUE) yoki «lampa o'chirilgan» (FALSE). Mantiqiy ma'lumotlarning informatikadagi ahamiyati beqiyos ekanligini informatika kursidan yaxshi bilasiz.

Ma'lumotlarni uch xil ko'rinishda ifodalash mumkin: ierarxik (shajaraviy), tarmoqli va relyatsion (jadvalli).

Bu turlarning har birini alohida ko'rib chiqamiz.

**Ierarxik tizim** jadvallardan tashkil topgan bo'lib, unda axborotni izlash jarayoni oldindan berilgan tartibda «ko'rib» chiqiladi. Ierarxik tizimda elementlarni joylashtirishda dasturlashgan tartib mavjud. Har bir guruhda biror element asosiy, qolganlari unga nisbatan ikkinchi darajali xususiyatga ega bo'ladi. Kerakli elementni tanlash ma'lum ketma-ketlikda olib boriladi.

Ma'lumotlarning daraxtsimon joylashishiga ma'lumotlarning **ierarxik modeli** deyiladi.

Ma'lumotlarning ierarxik modeliga mos holda yaratilgan ma'lumotlar omboriga **ma'lumotlarning ierarxik ombori** deyiladi.

Ma'lumotlarning ierarxik tizimini yaratish va undan foydalanish uchun yaratilgan tizimga **ma'lumotlar omborini boshqarishning ierarxik tizimi** deyiladi.

Ierarxik tizimli jadvalning birinchisida qator nomeri va unda joylashgan mahsulot kategoriyasi berilgan. Kategoriyalar keyingi sathda joylashgan mahsulot turini izlashda kerak bo'ladi. Ikkinchi va uchinchi satxdagi jadvalda esa har bir mahsulotning narxi haqidagi ma'lumot berilgan. Ma'lumotlar omborini ierarxik tizim tartibida tashkil qilish foydalanuvchi uchun ma'lum qulayliklar yaratadi.

**Tarmoqli tizim.** Tarmoqli tizim ierarxik tizimga nisbatan ancha qulayliklarga ega, chunki bu tizimda ma'lumot (fayl) lar o'rtasida ko'p yo'nalishli aloqalarni o'rnatish mumkin. Bunday aloqalar foydalanuvchiga kerakli ma'lumot (fayl)ni shajaraning yuqori sathlaridagi fayllarga murojaat qilmasdan topish imkonini beradi. Masalan, do'kondagi mahsulotlar jadvali bilan mahsulotlar narxi jadvallari orasida alohida bog'liqlik o'rnatilib, u kerakli mahsulot narxini topish imkonini beradi.

Bir pog'onadagi ma'lumotning boshqa pog'onadagi ma'lumot bilan ikki yoki undan ortiq marta bog'lanadigan turiga **ma'lumotlarning tarmoqli modeli** deyiladi.

Ma'lumotlarning tarmoqli tizimini yaratish va undan foydalanish uchun yaratilgan tizimga **ma'lumotlar omborini boshqarishning tarmoqli tizimi** deyiladi.

**Relyatsion tizim.** Relyatsion tizim 1970 yilda IBM firmasining xodimi E.F.Kodd tomonidan taklif etilgan bo'lib, hozirgi paytda eng ko'p tarqalgan tizimlar qatoriga kiradi. Chunki bu tizimda ma'lumotlar orasida eng qulay bog'lanishlarni amalga oshirish mumkin.

Ma'lumotlarning relyatsion modeliga mos holda yaratilgan ma'lumotlar omboriga **ma'lumotlarning relyatsion ombori** deyiladi.

Ma'lumotlarning relyatsion tizimini yaratish va undan foydalanish uchun yaratilgan tizimga **ma'lumotlar omborini boshqarishning relyatsion tizimi** deyiladi. Ma'lumotlarning jadval ko'rinishida saqlanishiga ma'lumotlarning **relyatsion modeli** deyiladi.

Relyatsion tizimdagi elementlarda shajaraning yo'qligi tufayli bu qulaylikka erishiladi. Axborotni izlashda barcha elementlardan kalit sifatida foydalanish mumkin. Ya'ni ixtiyoriy tuzilishga ega ma'lumotni ikki o'lchamli jadval ko'rinishiga keltirish mumkin.

Relyatsion tizim sifatida telefon ma'lumotnomasini tuzish masalasini ko'rib chiqamiz. Masalan, «har bir telefon nomeri ko'rsatilgan abonentning manzili, uning kategoriyasi belgilangan ro'yxatni tuzing» kabi o'qiladi.

### **Mustaqil ishlash uchun nazorat savollari:**

1. Ma'lumotlar strukturalarinini qanday turlari mavjud?
2. Ob'ekt nima?
3. Ob'ektlar to'plami nima? Ob'ekt nusxasi-chi?
4. Ma'lumotlarni boshqarish tizimi deganda nimani tushunasiz?
5. Sonli ma'lumotlar haqida gapirib bering.
6. Ma'lumotlarni necha xil ko'rinishda saqlash mumkin?

### **Mavzuga doir testlar:**

1. Ma'lumotlar deb nimaga aytiladi?
  - A) ma'lum bir shaklda qayd qilingan bo'lib, tahrirlash, uzatish uchun yaroqli bo'lgan voqelikka aytiladi
  - B) ma'lum bir shaklga ega bo'lmagan, qayta ishlash, saqlash va uzatish uchun yaroqli bo'lgan voqelikka aytiladi
  - C) ma'lum bir shaklda qayd qilingan bo'lib, qayta ishlash, saqlash va uzatish

- uchun yaroqli bo'lgan voqelikka aytiladi
- D) ma'lumotlar deb axborotning bir qismiga aytiladi
- E) barchasi to'g'ri
2. Ma'lumotlar ombori -
- A) jamiyatdagi axborotlar to'plamiga
- B) borliqni ma'lum bir sohasi bo'yicha birlashtirgan ma'lumotlar to'plami
- C) borliqdagi bir turdagi ma'lumotlar to'plami
- D) ma'lum bir soha bo'yicha kengaytirilgan axborotlar
- E) to'g'ri javob yo'q
3. Ma'lumotlarni uzatish va qabul qilishda qanday usuldan foydalaniladi?
- A) davolovchi va nazoratchi                      B) kriptografiya                      C) virus
- D) axborotni kodlash                      E) B va D
4. Relyatsion ma'lumotlar bazasi –
- A) moslanuvchanlik, aniqsizlik, maxfiylik, bog'liqsizlik, tushunarlilik
- B) soddalik, talabchanlik, maxfiylik, bog'liqlik, bog'liqsizlik
- C) moslanuvchanlik, tushunarlilik, aniqlilik, talabchanlik, maxfiylik
- D) soddalik, moslanuvchanlik, aniqlilik, maxfiylik, bog'liqlik, bog'liqsizlik
- E) barchasi to'g'ri
5. Ob'ekt –
- A) mavjud bo'lmagan va o'ziga o'xshash narsalardan ajralib turadigan alohida olingan predmet
- B) mavjud bo'lgan va o'ziga o'xshash narsalardan ajralib turadigan alohida olingan predmet
- C) mavjud bo'lgan va o'ziga o'xshamagan narsalardan ajralib turadigan alohida olingan predmet
- D) mavjud bo'lmagan va o'ziga o'xshamagan narsalardan ajralib turadigan alohida olingan predmet
- E) to'g'ri javob yo'q
6. Ma'lumotlarni modifikasiyalash –
- A) berilgan axborotni foydalanuvchi talabiga mos holda o'zgartirish
- B) axborotlarni foydalanuvchi talabiga mos holda nusxalash
- C) berilgan axborotni foydalanuvchi talabiga mos holda kuchirish
- D) axborotni foydalanuvchi talabiga mos holda uchirish
- E) to'g'ri javob yo'q

### **Adabiyotlar**

1. M.Aripov, A.Haydarov. Informatika asoslari. Toshkent. «O'qituvchi» 2002 yil. 297-302 betlar.
2. A.A.Abduqodirov, A.G'.Hayitov, R.R.Shodiyev. Axborot texnologiyalari. Toshkent. «O'qituvchi» 2003 yil, 36-49 betlar.



## 14 MA'RUZA: GRAFLAR. GRAFLAR ASOSIDA ALGORITMLAR.

### *Reja*

#### *1. Eng qisqa yo'llar masalalarining turlari.*

##### *1. Sozli algoritmni tuzish.*

##### *3. Algoritmni psevdokodda ishlab chiqish*

##### *4. Algoritmni baholash.*

**Darsning maqsadi:** Talabalarga graflar, graflar asosida algoritmlar? Graflarni boshqarish, optimallashtirish haqida ma'lumot berish.

**Tayanch iboralar:** Ma'lumotlar strukturasi, graf, yo'naltirilgan, simmetrik, asimmetrik.

**Mashg'ulot vositalari:** sinf doskasi, plakatlar, fundamental fan darsliklari, o'quv va uslubiy qo'llanmalar, informatika bo'yicha atamalar lug'ati, videoprojektor, ekran va kompyuterdan samarali foydalanish.

**Mashg'ulot usullari:** takrorlash, suhbat va savol-javob, munozara (mavzuni o'zlashtirishni mustahkamlash) tarzida muloqot o'tkazish, (talabalarning mustaqil, erkin fikrlash va so'zlashga o'rgatgan holda fikr mulohazalarini bayon qildirish, buning uchun har bir talabaga, tayanch iboralardan savollar tashlanadi, ular o'z fikrlarini bayon qiladilar, hamma talaba javobni bayon qilib bo'lgandan so'ng talaba bilan birgalikda javoblar yakun qilinadi).

**Darsning xronologik xaritasi** – 80 minut.

**Tashkiliy qismi:** Auditoriyaning jixozlanishi va sanitar sharoitlari, talabalar davomati – 2 minut.

**Bilimlarni baholash:** yangi mavzuni o'rganish uchun zarur bo'lgan material bo'yicha suxbat – 10 minut.

**Yangi mavzuni bayon etish** – 55 minut.

**Mavzu o'zlashtirilgan darajasini aniqlash** – 10 minut.

**Uyga vazifa** – 3 minut.

Yo'l tarmoqlari atlas (karta) qismi berilgan bo'lib, undan A va B nuqtalar orasidagi "eng yahshi" marshrutni topish kerak bo'lsin. "Eng yahshi" marshrutni ko'p faktorlar belgilashi mumkin, masalan, tezlik cheklangan holda marshrutni o'tish vaqti, o'tish kerak bo'lgan shaharlar soni va boshqalar.

Biz masalani eng qisqa yo'llar faktori bo'yicha yechamiz. Masalaning modeli turlar yordamida tuziladi. Uzlüksiz G turni har bir qirrasiga uning uzunligiga teng qiymat berilgan ko'rinishida tuzamiz. Bunday turda masofa irrallar yig'indisiga teng bo'ladi. Masalaning maqsadi ikkita berilgan uchlar orasidagi eng qisqa marshrutni topishdir.

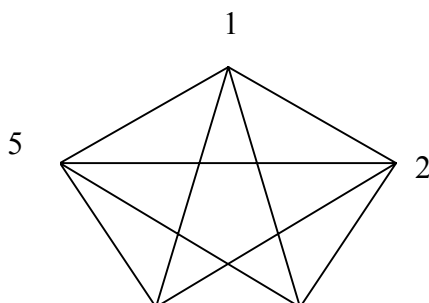
Umuman, eng qisqa yo'llar masalalari kombinator optimallashtirishning fundamental muammolaridandir. Ularning bir necha turlari mavjud, masalan, ikkita berilgan uchlar orasida, berilgan va qolgan barcha uchlar orasida, turdagi har bir uchlar juftliklari orasida va boshqalar.

Yo'l harajatlarini kamaytirish yechimini beradigan kommivoyajer Djek masalasini ko'ramiz. Djekning marshruti o'zi yashagan shahardan boshlanib, qolgan hamma shaharlarni bir martadan o'tib, yana o'z shahriga qaytib kelishi kerak. Demak, biz tuzayotgan ruyhatda har bir shahar faqat bir marta uchrashi kerak, Lekin Djek yashagan shahar ikki marta uchrab, ruyhatning birinchi va oxirgi elementlari bo'ladi. Undan tashqari, ruyhatdagi shaharlar tartibi Djekning marshrutini belgilaydi. Ruyhatdagi ikkita oxirgi shaharlar orasidagi yo'l narxi – bu butun marshrut narxi deb hisoblanadi. Demak, agar biz Djekga eng kichik narxdagi ruyhatni tuzib bersak, masalani yechgan bo'lamiz.

Masalani soddalashtirish uchun beshta shahar uchun yechim topamiz. Rasm. 1a – narxlar matrisasi. Rasm. 1b – turli model ko'rsatilgan.

-	1	2	7	5
1	-	4	4	3
2	4	-	1	2
7	4	1	-	3
5	3	2	3	-

Rasm 1-a). Narxlar matrisasi

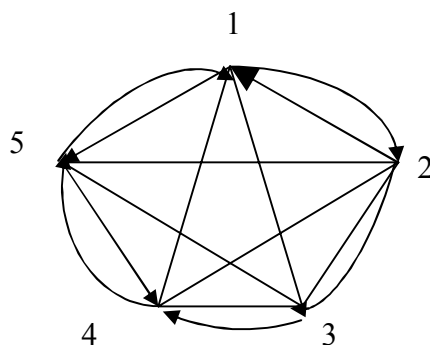


Rasm 1 4 5 o'rsimon mo 3

Masalani tarmoqlanish ko'rinishida taqqoslash qilamiz. Quyidagi rasmlarda beshta shahar uchun kommivoyajer assimetrik masalasining narxlar matrisasi berilgan.

	1	2	3	4	5
1	-	25	40	31	27
2	5	-	17	30	25
3	19	15	-	6	1
4	9	50	24	-	6
5	22	8	7	10	-

Rasm 1. Narxlar matrisasi



Rasm 2. O'rsimon mo 3

### Deykstra algoritmining so'zli tavsifi

Shunday masalalarni yechish uchun Deykstra algoritmi ancha qulay va yahshi deb topilgan. Algoritm quyidagi qadamlardan iborat:

1. Dastlab, berilgan **(Lex)** uchidan qolgan barcha uchlargacha bir qirra uzunligidagi masofalar aniqlanadi.
2. Ulardan eng qisqasi "doimiy eng qisqa masofa" sifatida belgilanadi (**Lex** va **BVa** uchlari qirradi).
3. Aniqlangan masofa **BVa** dan boshqa bor uchlargacha masofalarga qo'shiladi.
4. Hosil bo'lgan yig'indilar dastlab aniqlangan **Lex** dan qolgan uchlargacha bo'lgan masofalar bilan taqqoslanadi. Natijada masofasi qisqaroq bo'lgan uchning qirradi tanlanadi.

5. **BVa** uchi, eng qisqa masofa aniqlangan uch sifatida, ruyhatdan o'chiriladi. Ruyhatga boshqa uch qo'yiladi, masalan, **Roa. Bva** o'z navbatida, boshqa, izlanayotgan ruyhatga qo'yiladi.

Keyingi eng qisqa masofani topish uchun butun jarayon qayta bajariladi. **BVa** dan keyin yana bir uch ruyhatga qo'yiladi. Dastlabkisi esa ruyhatdan o'chiriladi. Sikl **Bed** va **Lex** uchlarini bog'lash uchun belgilangan qirralar aniqlanishi bilan to'xtatiladi.

Ko'rilgan misolda **Bed** uchi **Lex** dan boshlab eng oxirgi bo'lib chiqdi, ya'ni **Lex** dan **Bed** gacha eng qisqa masofani topish uchun biz **Lex** dan barcha qolgan uchlargacha eng qisqa yo'llarni topishga majbur bo'ldik.

Demak, eng yomon holatda 2 ta berilgan uchlar orasidagi eng qisqa yo'lni topish, bir berilgan nuqtadan qolgan barcha nuqtalargacha eng qisqa yo'l topish masalasi bilan murakkabligi bir xil bo'ladi.

### **Takrorlash ucun nazorat savollari**

1. *Qaysi mezonlar bo'yicha eng qisqa yo'llar masalalarini yechish mumkin?*
2. *Deykstra algoritmi nima uchun yaxshi hisoblanadi?*
3. *Algoritmni psevdokodda tavsiflashning qo'layligini ko'rsating*
4. *Deykstra algoritmining bahosi qanday?*

### **Mustaqil ishlash uchun nazorat savollari:**

1. Algoritmni baholash uchun qo'llanishi mumkin bo'lgan mezonlarni tavsiflab bering.
2. Vaqtli mezon bo'yicha baholash jarayoniga misollar ko'rsating.
3. Hajmiy mezon bo'yicha baholash jarayoniga misollar ko'rsating.
4. Eng qisqa yo'llarni topish masalasiga 3ta turli mezon bo'yicha yechim misollarini korsating.
5. Deykstra algoritmidan farqli boshqa eng qisqa yo'llarni topadigan algoritmi tuzing.

### **Mavzuga doir testlar:**

1. Quyida ikki algoritm keltirilgan:

1-algoritm: boshlanish  $i:=100, S1:=1$ ; toki  $i \geq 1$  takrorlash boshlanish  $S1:=S1+i; i:=i-1$  tamom; chikarish  $S1$ ; tamom.

2-algoritm: boshlanish  $i:=100, S2:=1$ ; toki  $i \geq 1$  takrorlash boshlanish  $S2:=S2*i; i:=i-1$  tamom; chikarish  $S2$ ; tamom.

Birinchi va ikkinchi algoritm bajarilishi natijasida mos ravishda  $S1$  va  $S2$  kiymatlar xosil kilinadi.  $S1$  va  $S2$  urtasida kuyidagi keltirilgan munosabatlardan kaysi biri bajariladi?

A)  $S1 < S2$ ;

B)  $S1 > S2$ ;

C)  $S1 = S2$ ;

D)  $S1 = 2 * S2$ ;

2. Obyektga yunaltirilgan dasturlashning asosiy goyasi?

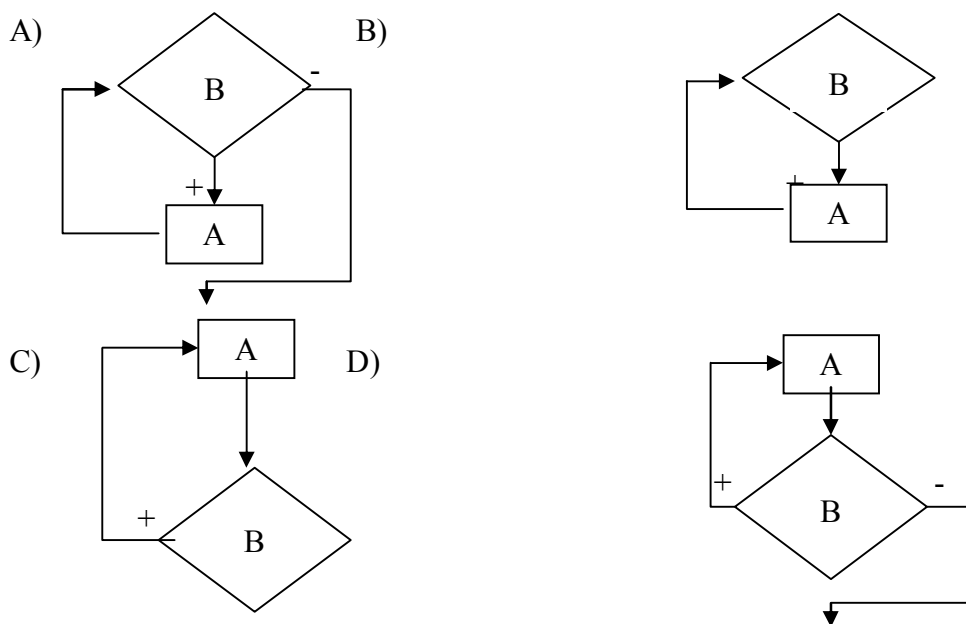
A) ma'lumotlar va ular ustida bajariladigan amallarni bir strukturaga birlashtirish;

B) ma'lumotlarni obyektlar sifatida tavsiflash;

C) ma'lumotlar va ular ustida bajariladigan amallarni aloxida-aloxida dasturlash;

D) obyektlar turi degan tushunchani kiritish

3. Old-shartli sikl operatori While B do A (bu yerda V-mantikiy turdagi ifoda, A-oddiy yoki murakkab operator)ning bajarilish jarayonini ifodalovchi blok-sxemani kursating.



4. Bir turdagi ma'lumotlar ketma-ketligini kompyuter xotirasida saqlash usuli shunday nomlanadi?

- A) Massiv
- B) Algoritm
- C) Šism-dastur
- D) Dastur

#### Adabiyotlar

1. В.А.Успенский, А.Л.Семенов. Теория алгоритмов: основные открытия и приложения. – М: Наука, 1987, 287 с.
2. Т.Кормен, Ч.Лейзерсон, Р.Ривест. Алгоритмы: построение и анализ. Сер: Классические учебники. М.: МЦНМО, 2001.- 960 с.
3. Гуломов С.С. ва бошқалар. Ахборот тизимлари ва технологиялари. Тошкент, 2000 й.
4. Жуманов И.И. Мингбаев Н.С., Информатика.- Самарқанд,: СамДУ нашри, 2002, 107 бет.
5. Ahatov A.R., Zaripova G.L. va boshq. Axborot texnologiyalari // Uslubiy qo'llanma. – Samarqand: SamDU nashri, 2008 yil – 112 bet.
6. Н. Вирт. Алгоритмы и структуры данных. – Досса, Хамарайан, 1997.
7. Жуманов И.И., Мингбоев Н.С. Ҳисоблаш системаларининг информацион асослари. Самарқанд: СамДУ нашри, 2002, 107 бет.

### 15 MA'RUZA: SHOXLAR VA CHEGARALAR ALGORITMI.

#### Reja

1. Masala qo'yilishi.
2. To'rsimon modellardan foydalanish.
3. Shoxlar bo'yicha baholash.
4. Chegaralar bo'yicha baholash.

**Darsning maqsadi:** Talabalarga graflar bilan berilgan masalani aniq algoritm misolida ko'rsatish. haqida ma'lumot berish. Algoritmni baholash va optimallashtirish bo'yicha kunikma hosil qilish

**Tayanch iboralar:** algoritmlar nazariyasi, evrisika, marshrut, minimum, maksimum, murakkablik, vaqti, hajmiy, mezon, chegara, optimallashtirish, test, hujaatlashtirish.

**Mashg'ulot vositalari:** sinf doskasi, plakatlar, fundamental fan darsliklari, o'quv va uslubiy qo'llanmalar, informatika bo'yicha atamalar lug'ati, videoprojektor, ekran va kompyuterdan samarali foydalanish.

**Mashg'ulot usullari:** takrorlash, suhbat va savol-javob, munozara (mavzuni o'zlashtirishni mustahkamlash) tarzida muloqot o'tkazish, (talabalarning mustaqil, erkin fikrlash va so'zlashga o'rgatgan holda fikr mulohazalarini bayon qildirish, buning uchun har bir talabaga, tayanch iboralardan savollar tashlanadi, ular o'z fikrlarini bayon qiladilar, hamma talaba javobni bayon qilib bo'lgandan so'ng talaba bilan birgalikda javoblar yakun qilinadi).

**Darsning xronologik xaritasi** – 80 minut.

**Tashkiliy qismi:** Auditoriyaning jixozlanishi va sanitar sharoitlari, talabalar davomati – 2 minut.

**Bilimlarni baholash:** yangi mavzuni o'rganish uchun zarur bo'lgan material bo'yicha suxbat – 10 minut.

**Yangi mavzuni bayon etish** – 55 minut.

**Mavzu o'zlashtirilgan darajasini aniqlash** – 10 minut.

**Uyga vazifa** – 3 minut.

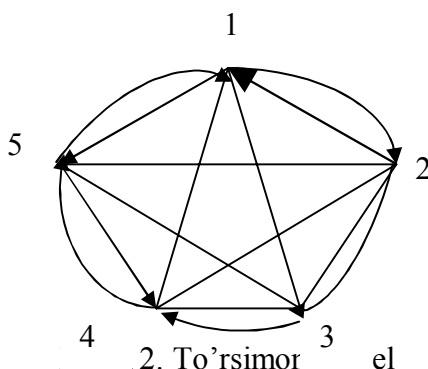
Bu usul yechimlar fazosining tursimon modelini ta'qiq qiladigan usullar turiga kiradi va kombinatorika masalalarining keng doirasiga qo'llanilishi mumkin.

Bunday algoritmlar ko'proq optimizatsiyaga yo'naltirilgan va ancha murakkab bo'ladi, lekin kommivoyajer masalasini yechishda juda qulay hisoblanadi.

Masalani tarmoqlanish ko'rinishida tadqiq qilamiz. Quyidagi rasmlarda beshta shahar uchun kommivoyajer assimetrik masalasining narxlar matrisasi berilgan.

	1	2	3	4	5
1	-	25	40	31	27
2	5	-	17	30	25
3	19	15	-	6	1
4	9	50	24	-	6
5	22	8	7	10	-

Rasm 1. Narxlar matrisasi



Bundan tashqari rasmda narxlarni ko'rsatish uchun yo'naltirilgan tarmoqdan foydalanamiz. Bu yerda  $i$  shahardan  $j$  shaharga borish bahosi,  $j$  dan  $i$  ga borish bahosiga teng bo'lishi shart emas. Bizning izlash daraxtimizning ildizi barcha mumkin bo'lgan marshrutlar to'plamiga mos bo'ladi, ya'ni besh shahar masalasidagi (4!) marshrutlar to'plamini aks ettiradi. Umuman, ixtiyoriy  $N$  shaharni assimetrik masala uchun ildiz barcha  $\{(N-1)!\}$  mumkin bo'lgan marshrutlar  $R$  to'plamini akslantiradi. Ildizdan tarqaladigan shohlar bir qirrani, masalan,  $(i,j)$  – ni tanlash bilan aniqlanadi. Bu ishdan maqsad – barcha marshrutlar to'plamini ikki to'plamga ajratish: Biri optimallashtirilgan tur, ikkinchisi esa optimallashtirilmagan turlardan iborat bo'ladi.  $(i,j)$  tanlangan qirra optimal turga tegishli deb hisoblagan holda,  $R$  to'plamni ikkiga bo'lamiz, ya'ni  $\{i,j\}$  va  $\{\bar{i},\bar{j}\}$  to'plamlarga.  $\{i,j\}$  to'plamiga  $(i,j)$  qirrasini qatnashgan turlar kiradi,  $\{\bar{i},\bar{j}\}$  to'plamga esa shu qirra qatnashmagan tur.

Faraz qilaylik, biz tarmoqlanishni  $\{i,j\}=\{3,5\}$  qirrasida amalga oshirdik, chunki bu qirraning bahosi matrisada eng kichik. Unda rasmda ildiz va uning birinchi darajasini ko'rsatishimiz mumkin.

Shuni ta'kidlash kerakki,  $R$ -ga tegishli har bir tur birinchi darajaning faqatgina bitta to'plamiga kiradi. Agar biz  $\{3,5\}$  to'plamida optimal tur yo'qligini qabul qilsak,  $\{3,5\}$  to'plamini tadqiq qilishga o'tamiz.  $\{3,5\}$  to'plamini ham yuqoridagidek bo'lamiz. Arzonlik bo'yicha (2,1) qirrasini matrisada ikkinchi o'rinda  $C(2,1)=5$ . Shuning uchun  $\{3,5\}$  to'plamini  $Y$  va  $\bar{Y}$  deb belgilaymiz.  $Y$  to'plamga  $X$  to'plamda qatnashgan va  $(i,j)$  qirrasini mavjud turlar kiradi,  $\bar{Y}$  to'plamga  $(i,j)$  qirrasini qatnashmagan  $X$  ning qism to'plami.

Yuqorida tadqiq qilingan jarayon tarmoqlanish haqida tasavvur beradi. Endi chegaralar hisoblashni ko'ramiz.

Har bir daraxt uchi bilan shu uch bilan belgilangan to'plamning ixtiyoriy turining pastki narx chegarasini bog'laymiz. Bunday chegaralarni hisoblash – shohlar va chegaralar kabi usullarda tadqiqotlarni yengillashtirish uchun asosiy faktordir. Shuning uchun ularni aniq hisoblashga katta e'tibor berish lozim.

Sababi quyidagicha: Masalan,  $m$  baholi konkret bir turni qabul qilaylik. Unda, agar  $V_k$  uchi bilan belgilangan turlar to'plami bilan bog'liq pastki chegara  $M \geq m$  bo'lsa, optimal turni izlash jarayoni davomida  $V_k$  va undan keyingi uchlarni tadqiq qilish kerak bo'lmay qoladi.

Xulosa qilib, shuni aytish mumkin-ki, shohlar va chegaralar uslubi murakkab bo'lsa-da, kommivoyajer masalasi katta sonli shaharlar va narxlar bilan berilganda, algoritmlar aniq va tez ishlaydi, algoritmlarning murakkabligi esa eksponensial.

### **Takrorlash ucun savollar**

1. Kommivoyajer masalasida ikki tomonli narxlar matrisasi qaysi holatda tuziladi.
2. To'rsimon modellardan foydalanish.
3. Shohlar bo'yicha baholashning afzalligini tushuntirib bering.
4. Chegaralar bo'yicha baholash nimadan iborat?

### **Mustaqil ishlash uchun nazorat savollari:**

6. Algoritmni baholash uchun qo'llanishi mumkin bo'lgan mezonlarni tavsiflab bering.
7. Vaqtli mezon bo'yicha baholash jarayoniga misollar ko'rsating.
8. Hajmiy mezon bo'yicha baholash jarayoniga misollar ko'rsating.
9. Eng qisqa yo'llarni topish masalasiga 3ta turli mezon bo'yicha yechim misollarini korsating.
10. Deykstra algoritmidan farqli boshqa eng qisqa yo'llarni topadigan algoritmni tuzing.

### Adabiyotlar

1. Леонтьев В.. Новейшая энциклопедия персонального компьютера. - М.: Олма пресс образование, Москва. -2005.
2. Qobulov V.Q. Aql mo'jizasi. - T.,: Fan, Toshkent. - 1984.
3. Jumanov I.I., Mingboyev N.S.. Informatika. Uslubiy qo'llanma. – Samarqand: SamDU. - 2002.
4. Nurmammedov T.A. IBM PC va MS DOS bilan ishlash. - T.: Fan, Toshkent – 1995.
5. G'ulomov S.S., Shermammedov A.T., Begalov B.A. Iqtisodiy informatika. – T.: “O'zbekiston”, Toshkent. – 1999.
6. Бройдо В.Л. Ofis texnikasi (boshqarish va ish yuritish uchun). – T.: Mehnat, Toshkent. - 2001.
7. Qobilov S.S., Jumanov I.I. SUBD i informasionniye sistemi. – Samarqand: SamDU. - 1997.
8. Aripov M. Internet va elektron pochta aloqasi. - T.: «Universitet».- 2000.
9. Jumanov I.I., Mingboyev N.S. Hisoblash sistemalarining informatsion asoslari. – Samarqand: SamDU. – 2002.
10. G'ulomov S.S. va boshqalar. «Iqtisodiy informatika». - T.: Fan - 1999.
11. Raxmonqulova S.I. IBM PC shaxsiy kompyuterida ishlash. - T.: Fan, Toshkent – 1999.
12. Nasretdinova Sh. Windows uchun Excel sahifalarida. - T.: Fan. – 1999.
13. Фигурнов В.Э. IBM PC для пользователя. - М.: Инфра, 1996.
14. Шафрин Ю. Основы компьютерной технологии. - Б.: Туркистон, Бишкек. – 1998.