

## 1-Ma'ruza.

### 1-Mavzu. Qurilish va arxitektura soxasida axborot jarayonlarini algoritmlash va dasturlash

#### Reja:

1. Algoritm tushunchasi.
2. Algoritmning xossalari.
3. Algoritmning turlari.
4. Algoritmning tasvirlash usullari.

**Tayanch so'zlar:** *Algoritm, diskretlilik, tushunarлilik, aniqlik, ommaviylik, natijaviylik, blok-sxemalar, tarmoqlanuvchi algoritmlar, takrorlanuvchi algoritmlar, siklik algoritmlar.*

Algoritm bu oldimizga qo'yilgan masalani yechish uchun zarur bo'lgan amallar ketma-ketligidir.

Algoritm so'zi va tushunchasi IX asrda yashab ijod etgan buyur alloma Muhammad al-Xorazmiy nomi bilan uzviy bog'liq. Algoritm so'zi Al-Xorazmiy nomini Yevropa olimlari tomonidan buzib talaffuz qilinishidan yuzaga kelgan. Al-Xorazmiy birinchi bo'lib o'nlik sanoq sistemasining tamoyillarini va undagi to'rtta amallarni bajarish qoidalarini asoslab bergan.

**Algoritmning asosiy xossalari.** Algoritmning 5-ta asosiy xossasi bor: **Diskretlilik (Cheklilik).** Bu xossaning mazmuni algoritmlarni doimo chekli qadamlardan iborat qilib bo'laklash imkoniyati mavjudligida. Ya'ni uni chekli sondagi oddiy ko'rsatmalar ketma-ketligi shaklida ifodalash mumkin. Agar kuzatilayotgan jarayonni chekli qadamlardan iborat qilib qo'llay olmasak, uni algoritm deb bo'lmaydi.

**Tushunarлilik.** Biz kundalik hayotimizda berilgan algoritmlar bilan ishlayotgan elektron soatlar, mashinalar, dastgohlar, kompyuterlar, turli avtomatik va mexanik qurilmalarni kuzatamiz.

Ijrochiga tavsiya etilayotgan ko'rsatmalar, uning uchun tushinarli mazmunda bo'lishi shart, aks holda ijrochi oddiygina amalni ham bajara olmaydi. Undan tashqari, ijrochi har qanday amalni bajara olmasligi ham mumkin.

Har bir ijrochining bajarishi mumkin bo'lgan ko'rsatmalar yoki buyruqlar majmuasi mavjud, u ijrochining ko'rsatmalar tizimi (sistemasi) deyiladi. Demak, ijrochi uchun berilayotgan har bir ko'rsatma ijrochining ko'rsatmalar tizimiga mansub bo'lishi lozim.

Ko'rsatmalarni ijrochining ko'rsatmalar tizimiga tegishli bo'ladigan qilib ifodalay bilishimiz muhim ahamiyatga ega. Masalan, quyi sinfning a'lochi o'quvchisi "son kvadratga oshirilsin" degan ko'rsatmani tushinmasligi natijasida bajara olmaydi, lekin "son o'zini o'ziga ko'paytirilsin" shaklidagi ko'rsatmani bemalol bajaradi, chunki u ko'rsatma mazmunidan ko'paytirish amalini bajarish kerakligini anglaydi.

**Aniqlik.** Ijrochiga berilayotgan ko'rsatmalar aniq mazmunda bo'lishi zarur. Chunki ko'rsatmadagi noaniqliklar mo'ljaldagi maqsadga erishishga olib kelmaydi. Odam uchun tushinarli bo'lgan "3-4 marta silkitilsin", "5-10 daqiqa qizdirilsin", "1-2 qoshiq solinsin", "tenglamalardan biri yechilsin" kabi noaniq ko'rsatmalar robot yoki kompyuterni qiyin ahvolga solib qo'yadi.

Bundan tashqari, ko'rsatmalarining qaysi ketma-ketlikda bajarilishi ham muhim ahamiyatga ega. Demak, ko'rsatmalar aniq berilishi va faqat algoritmda ko'rsatilgan tartibda bajarilishi shart ekan.

**Ommaviylik.** Har bir algoritm mazmuniga ko'ra bir turdag'i masalalarning barchasi uchun ham o'rinni bo'lishi kerak. YA'ni masaladagi boshlang'ich ma'lumotlar qanday bo'lishidan qat'iy nazar algoritm shu xildagi har qanday masalani yechishga yaroqli bo'lishi kerak. Masalan, ikki oddiy kasrning umumiyligi mahrajini topish algoritmi, kasrlarni turlicha o'zgartirib bersangiz ham ularning umumiyligi mahrajlarini aniqlab beraveradi. Yoki uchburchakning yuzini topish algoritmi, uchburchakning qanday bo'lishidan qat'iy nazar, uning yuzini hisoblab beraveradi.

**Natijaviylik.** Har bir algoritm chekli sondagi qadamlardan so‘ng albatta natija berishi shart. Bajariladigan amallar ko‘p bo‘lsa ham baribir natijaga olib kelishi kerak. Chekli qadamdan so‘ng qo‘yilgan masala yechimga ega emasligini aniqlash ham natija hisoblanadi. Agar ko‘rilayotgan jarayon cheksiz davom etib natija bermasa, uni algoritm deb atay olmaymiz.

**Algoritmning tasvirlash usullari**. Yuqorida ko‘rilgan *misollarda* odatda biz masalani yechish algoritmini so‘zlar va matematik formulalar orqali ifodaladik. Lekin algoritm boshqa ko‘rinishlarda ham berilishi mumkin. Biz endi algoritmlarning eng ko‘p uchraydigan turlari bilan tanishamiz.

**1. Algoritmning so‘zlar orqali ifodalanishi.** Bu usulda ijrochi uchun beriladigan har bir ko‘rsatma jumlalar, so‘zlar orqali buyruq shaklida beriladi.

**2. Algoritmning formulalar bilan berilish** usulidan matematika, fizika, kimyo kabi aniq fanlardagi formulalarni o‘rganishda foydalaniladi. Bu usulni ba’zan analitik ifodalash deyiladi.

**3. Algoritmlarning grafik shaklida tasvirlanishida** algoritmlar maxsus geometrik figuralar yordamida tasvirlanadi va bu grafik ko‘rinishi blok-sxema deyiladi.

**4. Algoritmning jadval ko‘rinishda berilishi.** Algoritmning bu tarzda tasvirlanishdan ham ko‘p foydalanamiz. Masalan, maktabda qo‘llanib kelinayotgan to‘rt xonali matematik jadvallar yoki turli xil lotereyalar jadvallari. Funksiyalarning grafiklarini chizishda ham algoritmlarning qiymatlari jadvali ko‘rinishlaridan foydalanamiz. Bu kabi jadvallardan foydalanish algoritmlari sodda bo‘lgan tufayli ularni o‘zlashtirib olish oson.

Yuqorida ko‘rilgan algoritmlarning tasvirlash usullarining asosiy maqsadi, qo‘yilgan masalani yechish uchun zarur bo‘lgan amallar ketma-ketligining eng qulay holatinni aniqlash va shu bilan odam tomonidan programma yozishni yanada osonlashtirishdan iborat. Aslida programma ham algoritmning boshqa bir ko‘rinishi bo‘lib, u insonning kompyuter bilan muloqotini qulayroq amalga oshirish uchun mo‘ljallangan.

Blok-sxemalarni tuzishda foydalaniladigan asosiy sodda geometrik figuralar quyidagilardan iborat 1-Jadval.

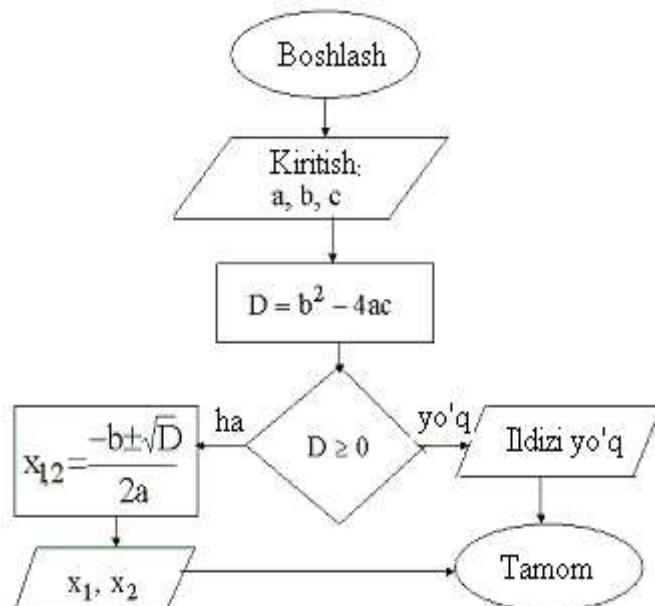
Nomi	Belgilanishi	Bajaradigan vazifasi
Jarayon		Bir yoki bir nechta amallarni bajarilishi natijasida ma’lumotlarning uzgarishi
Qaror		Biror shartga boglik ravishda algoritmning bajarilish yo’nalishini tanlash
SHakl uzgartirish		Dasturni uzgartiruvchi buyruk yoki buyruklar turkumini uzgartirish amalini bajarish
Avval aniqlangan jarayon		Oldindan ishlab chikilgan dastur yoki algoritmdan foydalanish
Kiritish Chiqarish		Axborotlarni kayta ishlash mumkin bo‘lgan shaklga utkazish yoki olingan natijani tasvirlash

Display		kompyuterga ulagan displaydan axborotlarni kiritish yoki chiqarish
Hujjat		Axborotlarni kogozga chiqarish yoki kogozdan kiritish

Blok-sxemalar bilan ishlashni yaxshilab o'zlashtirib olish zarur, chunki bu usul algoritmlarni ifodalashning qulay vositalaridan biri bo'lib programma tuzishni osonlashtiradi, programmalash qobiliyatini mustahkamlaydi. Algoritmik tillarda blok - sxemaning asosiy strukturalariga maxsus operatorlar mos keladi.

Shuni aytish kerakni, blok-sxemalardagi yozuvlar odatdagи yozuvlardan katta farq qilmaydi.

Misol sifatida  $ax^2+bx+c=0$  kvadrat tenglamani yechish algoritmining blok-sxemasini quyida keltirilgan.



Kvadrat tenglamani yechish algoritmi chiziqli algoritmlar.

Har qanday murakkab algoritmnin ham uchta asosiy struktura yordamida tasvirlash mumkin. Bular ketma-ketlik, ayri va takrorlash strukturalaridir. Bu strukturalar asosida chiziqli, tarmoqlanuvchi va takrorlanuvchi hisoblash jarayonlarining algoritmlarini tuzish mumkin. Umuman olganda, algoritmlarni shartli ravishda quyidagi turlarga ajratish mumkin:

chiziqli algoritmlar;

tarmoqlanuvchi algoritmlar;

takrorlanuvchi yoki siklik algoritmlar;

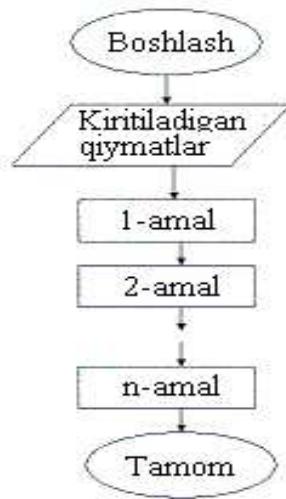
ichma-ich joylashgan siklik algoritmlar;

rekurrent algoritmlar;

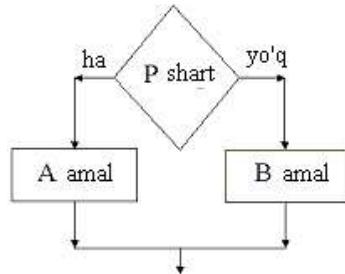
takrorlanishlar soni oldindan no'malum algoritmlar;

ketma-ket yaqinlashuvchi algoritmlar.

Faqat ketma-ket bajariladigan amallardan tashkil topgan algoritmlarga-chiziqli algoritmlar deyiladi. Bunday algoritmnin ifodalash uchun ketma-ketlik strukturasi ishlatiladi. Strukturada bajariladigan amal mos keluvchi shakl bilan ko'rsatiladi. Chiziqli algoritmlar blok-sxemasining umumiy strukturasini quyidagi ko'rinishda ifodalash mumkin:



**Tarmoqlanuvchi algoritmlar.** Agar hisoblash jarayoni biror bir berilgan shartning bajarilishiga qarab turli tarmoqlar bo'yicha davom ettirilsa va hisoblash jarayonida har bir tarmoq faqat bir marta bajarilsa, bunday hisoblash jarayonlariga tarmoqlanuvchi algoritmlar deyiladi. Tarmoqlanuvchi algoritmlar uchun ayri strukturasi ishlataladi. Tarmoqlanuvchi strukturasi berilgan shartning bajarilishiga qarab ko'rsatilgan tarmoqdan faqat bittasining bajarilishini ta'minlaydi.



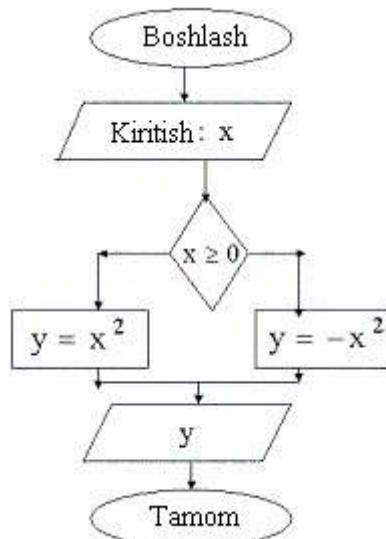
#### Tarmoqlanishning umumiy ko'rinishi

Berilgan shart romb orqali ifodalanadi, r-berilgan shart. Agar shart bajarilsa, "ha" tarmoq bo'yicha a amal, shart bajarilmasa "yo'q" tarmoq bo'yicha b amal bajariladi.

Tarmoqlanuvchi algoritmgaga tipik *misol* sifatida quyidagi sodda *misolni* qaraylik.

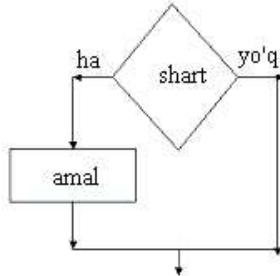
$$Y = \begin{cases} x^2 & \text{agar } x \geq 0 \\ -x^2 & \text{agar } x < 0 \end{cases}$$

Berilgan  $x$  ning qiymatiga bog'lik holda, agar u musbat bo'lsa «ha» tarmoq bo'yicha  $y=x^2$  funksiyaning qiymati, aks holda  $y=-x^2$  funksiyaning qiymati hisoblanadi.



Interval ko‘rinishidagi funksiya qiyMatni hisoblash algoritmi

Ko‘pgina masalalarni yechishda, shart asosida tarmoqlanuvchi algoritmlarning ikkita tarmog‘idan bittasining, ya’ni yoki «ha» yoki «yo‘q» ning bajarilishi yetarli bo‘ladi. Bu holat tarmoqlanuvchi algoritmnинг xususiy holi sifatida aylanish strukturasi deb atash mumkin. Aylanish strukturasi quyidagi ko‘rinishga ega:



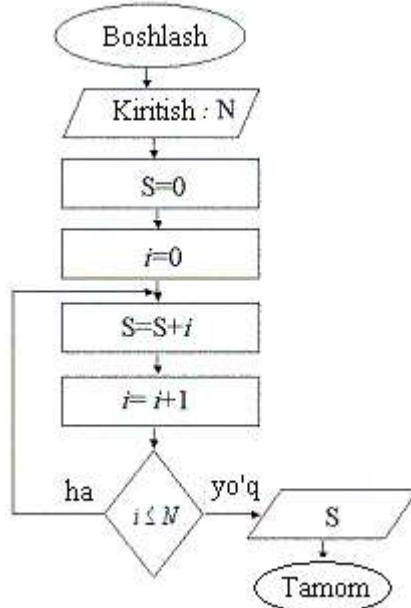
Aylanish strukturasining umumiy ko‘rinishi

**Takrorlanuvchi algoritmlar**. Agar biror masalani yechish uchun tuzilgan zarur bo‘lgan amallar ketma-ketligining ma’lum bir qismi biror parametriga bog‘liq ko‘p marta qayta bajarilsa, bunday algoritm takrorlanuvchi algoritm yoki siklik algoritmlar deyiladi. Takrorlanuvchi algoritmlarga tipik *misol* sifatida odatda qatorlarning yig‘indisi yoki ko‘patmasini hisoblash jarayonlarini qarash mumkin. Quyidagi yig‘indini hisoblash algoritmini tuzaylik.

$$S = 1 + 2 + 3 + \dots + N = \sum_{i=1}^N i$$

Bu yig‘indini hisoblash uchun  $i=0$  da  $S=0$  deb olamiz va  $i=i+1$  da  $S=S+i$  ni hisoblaymiz. Bu yerda birinchi va ikkinchi qadamlar uchun yig‘indi hisoblandi va keyingi qadamda  $i$  parametr yana bittaga orttiriladi va navbatdagi raqam avvalgi yig‘indi  $S$  ning ustiga qo‘shiladi va bu jarayon shu tartibda to  $i < N$  sharti bajarilmaguncha davom ettiriladi va natijada izlangan yig‘indiga ega bo‘lamiz. Bu fikrlarni quyidagi algoritmda ifodalash mumkin:

$N$  –berilgan bo‘lsin,  $i=0$  berilsin,  $S=0$  berilsin,  $i=i+1$  hisoblansin,  $S=S+i$  hisoblansin,  $i < N$  tekshirilsin va bu shart bajarilsa, 4- satrga qaytilsin, aks holda keyingi qatorga o‘tilsin,  $S$  ning qiymati chop etilsin.

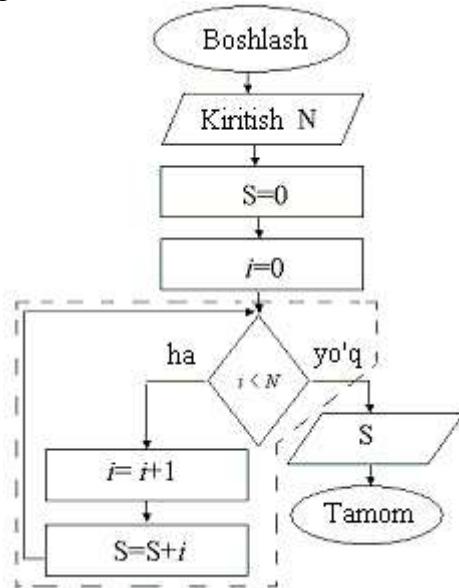


1 dan  $n$  gacha bo‘lgan sonlar yig‘indisini hisoblash algoritmi

Yuqorida keltirilgan algoritm va blok sxemadan ko‘rinib turibdiki amallar ketma-ketligining ma’lum qismi parametr  $i$  ga nisbatan  $N$  marta takrorlanayapti.

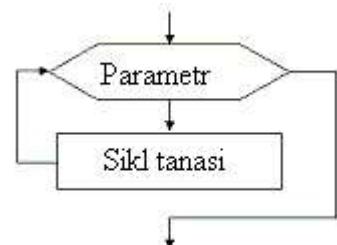
Yuqorida ko‘rilgan yig‘indi blok sxemalaridagi takrorlanuvchi qismlariga (aylana ichiga olingan) quyidagi sharti keyin berilgan siklik struktura mos kelishini ko‘rish mumkin.

Yuqoridagi blok sxemalarda shartni oldin tekshiriladigan holatda chizish mumkin edi. Masalan, yig'indining algoritmini qaraylik. Bu blok sxemaning takrorlanuvchi qismiga quyidagi, sharti oldin berilgan siklik strukturaning mos kelishini ko'rish mumkin.



1 dan  $n$  gacha bo'lgan sonlar yig'indisini hisoblash algoritmi

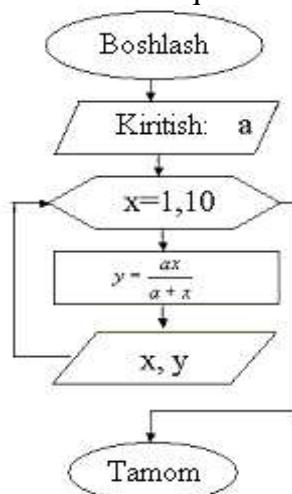
Blok sxemalarining takrorlanuvchi qismlarini, quyidagi parametrli takrorlash strukturasini ko'rinishida ham ifodalash mumkin.



Parametrli takrorlash operatorining umumiy ko'rinishi

$$y = \frac{ax}{a+x}$$

Parametrli takrorlash operatoriga *misol* sifatida berilgan  $x=1,2,3,\dots,10$  larda funksiyasining qiymatlarini hisoblash blok sxemasini qarash mumkin.



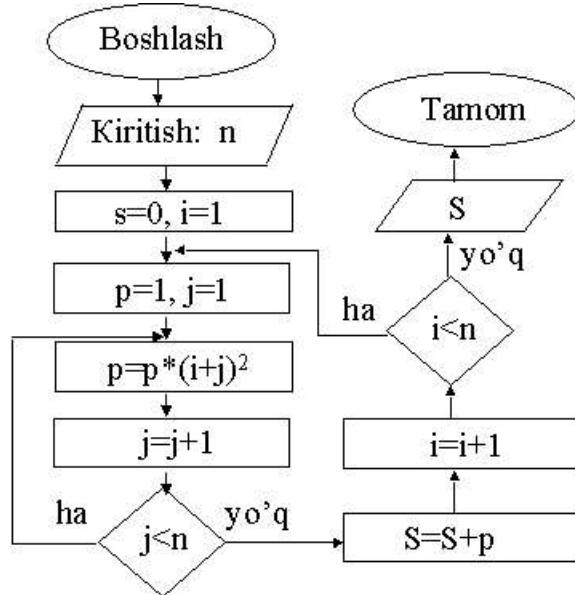
Parametrli takrorlash operatoriga doir algoritmi

**Ichma-ich joylashgan siklik algoritmlar**. Ba'zan, takrorlanuvchi algoritmlar bir nechta parametrarga bog'liq bo'ladi. Odatda bunday algoritmlarni ichma-ich joylashgan algortmlar deb ataladi.

Misol sifati berilgan  $n \times m$  o'lchovli  $a_{ij}$  -matritsa elementlarining yig'indisini hisoblash masalasini qaraylik.

$$S = \sum_{i=1}^n \prod_{j=1}^m (i+j)^2$$

Bu yig'indi hisoblash uchun,  $i$  ning har bir qiymatida  $j$  bo'yicha ko'paytmani hisoblab, avval yig'indi ustiga ketma-ket qo'shib borish kerak bo'ladi. Bu jarayon quyidagi blok-sxemada aks ettirilgan. Bu yerda  $i$ -tashqi sikl - yig'indi uchun,  $j$ -esa ichki sikl-ko'paytmani hosil qilish uchun foydalanilgan.



Ichma-ich joylashgan siklik algoritmga doir blok-sxema