

Andijon Davlat Universiteti fizika-matematika fakulteti
Informatika yo'nalishi III-bosqich IAT guruh talabasi
Qambarova Nodiraningning
“Algoritmlar, ularning xossalari.
Berilish usullari va strukturalari
” Mavzusidagi

REFERATI

Mavzu. Algoritmlar, ularning xossalari. Berilish usullari va strukturalari

- 1. Reja:**
- 2. Algoritmning asosiy xossalari**
- 3. Algoritmning tasvirlash usullari**
- 4. Chiziqli algoritmlar**
- 5. Tarmoqlanuvchi algoritmlar**
- 6. Takrorlanuvchi algoritmlar**
- 7. Algoritm ijrosini tekshirish**

Kalit so'zlar: Aniqlik, Diskretlik, Ommaviylik, Tushunarlilik, Natijaviylik, Blok-sxema, Algoritmik notasiya

Yuqorida qayd qilganimizdek, qo'yilgan biror masalani EHMda yechish uchun, avval uning matematik modelini, keyin algoritmini va programmasini tuzish kerak bo'ladi. Bu uchlikda algoritm bloki muhim ahamiyatga ega. Endi algoritm tushunchasining ta'rif va xossalarini bayon qilamiz.

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

Algoritm so'zi va tushunchasi IX asrda yashab ijod etgan buyuk 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.

Tushunarlilik. 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 (sistemi) 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'rsatmalarning 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 turdagi masalalarning barchasi uchun ham o'rinli 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 umumiy mahrajini topish algoritmi, kasrlarni turlicha o'zgartirib bersangiz ham ularning umumiy 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 algoritmning eng ko'p uchraydigan turlari bilan tanishamiz.

Algoritmning so'zlar orqali ifodalanishi. Bu usulda ijrochi uchun beriladigan har bir ko'rsatma jumlar, so'zlar orqali buyruq shaklida beriladi.

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

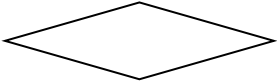
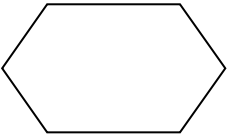
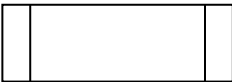
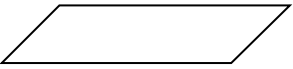
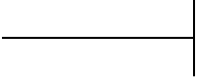
3. Algoritmning grafik shaklida tasvirlanishida algoritmlar maxsus geometrik figuralar yordamida tasvirlanadi va bu grafik ko'rinishi blok-sxema deyiladi.

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 algoritmning qiymatlari jadvali ko'rinishlaridan foydalanamiz. Bu kabi jadvallardan foydalanish algoritmlari sodda bo'lgan tufayli ularni o'zlashtirib olish oson.

Yuqorida ko'rilgan algoritmning tasvirlash usullarining asosiy maqsadi, qo'yilgan masalani yechish uchun zarur bo'lgan amallar ketma-ketligining eng qulay holatini 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.

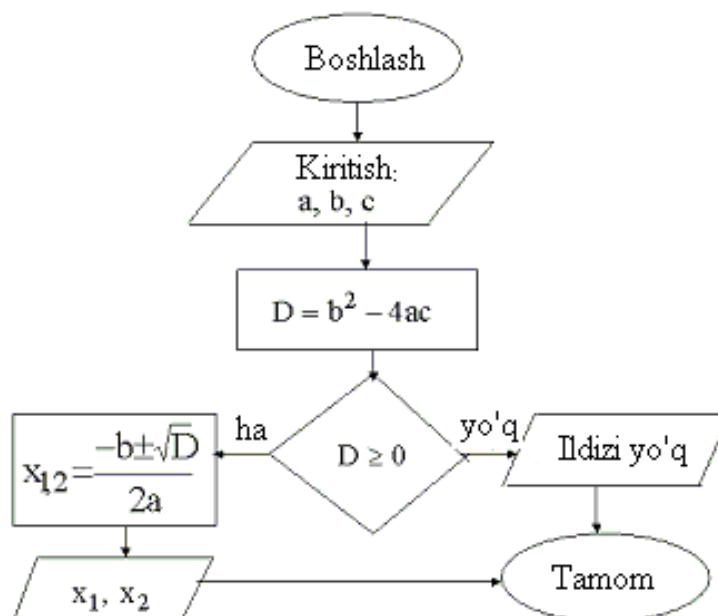
Blok-sxemalarni tuzishda foydalaniladigan asosiy sodda geometrik figuralar quyidagilardan iborat:

Nomi	Belgilanishi	Bajaradigan vazifasi
Jarayon		Bir yoki bir nechta amallarni bajarilishi natijasida ma'lumotlarning uzgarishi
Karor		Biror shartga bog'liq ravishda algoritmning bajarilish yunalishini tanlash
SHakl uzgartirish		Dasturni uzgartiruvchi buyruk yoki buyruklar turkumini uzgartirish amalini bajarish
Avval aniklangan jarayon		Oldindan ishlab chikilgan dastur yoki algoritmdan foydalanish
Kiritish CHikarish		Axborotlarni kayta ishlash mumkin bulgan shaklga utkazish yoki olingan natijani tasvirlash
Displey		EXMga ulangan displeydan axborotlarni kiritish yoki chikarish
Xujjat		Axborotlarni kogoza chikarish yoki kogoza kiritish
Axborotlar okimi chizigi		Bloklar orasidagi boglanishlarni tasvirlash
Boglagich		Uzilib kolgan axborot okimlarini ulash belgisi
Boshlash Tugatish		Axborotni kayta ishlashni boshlash, vaktincha yoki butunlay tuxtatish
Izox		Bloklarga tegishli turli xildagi tushuntirishlar

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 odatdagi yozuvlardan katta farq qilmaydi.

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

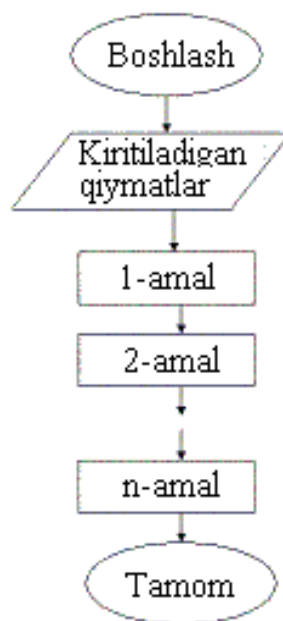


1-rasm. Kvadrat tenglamani yechish algoritmi

Chiziqli algoritmlar. Har qanday murakkab algoritmnı 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:

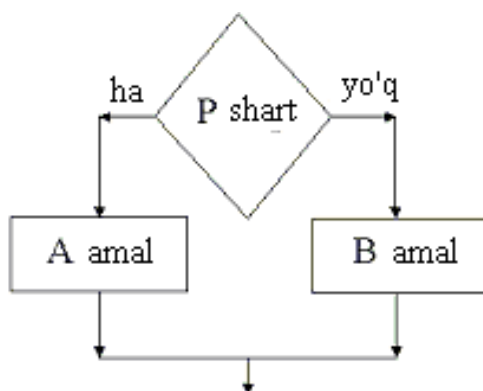
- chiziqli algoritmlar;
- tarmoqlanuvchi algoritmlar;
- takrorlanuvchi yoki siklik algoritmlar;
- ichma-ich joylashgan siklik algoritmlar;
- rekurrent algoritmlar;
- takrorlanishlar soni oldindan no'malum algoritmlar;
- ketma-ket yaqinlashuvchi algoritmlar.

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



2-rasm. Chiziqli algoritmlar blok - sxemasining umumiy strukturalari

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 strukturalari ishlatiladi. Tarmoqlanuvchi strukturalari berilgan shartning bajarilishiga qarab ko'rsatilgan tarmoqdan faqat bittasining bajarilishini ta'minlaydi.



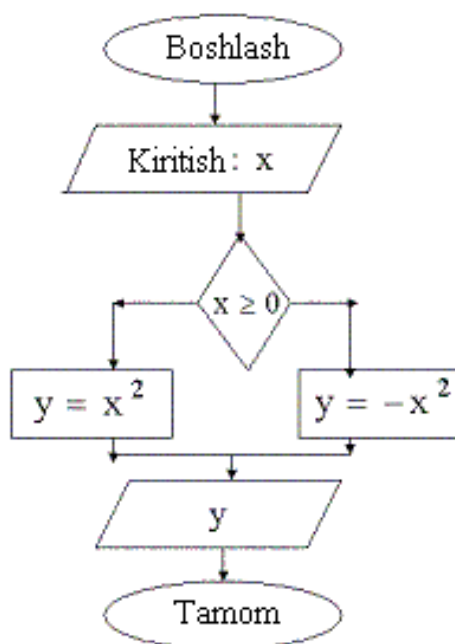
3-rasm. Tarmoqlanishning umumiy ko'rinishi

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 algoritimga tipik *misol* sifatida quyidagi sodda *misol*ni qaraylik.

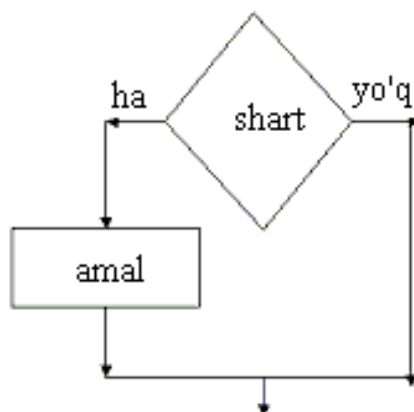
1- Misol:
$$Y = \begin{cases} x^2 & \text{agar } x \geq 0 \\ -x^2 & \text{agar } x < 0 \end{cases}$$

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.



4-rasm. Interval ko'rinishidagi funksiya qiymatini 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 algoritmning xususiy holi sifatida aylanish strukturasi deb atash mumkin. Aylanish strukturasi quyidagi ko'rinishga ega:



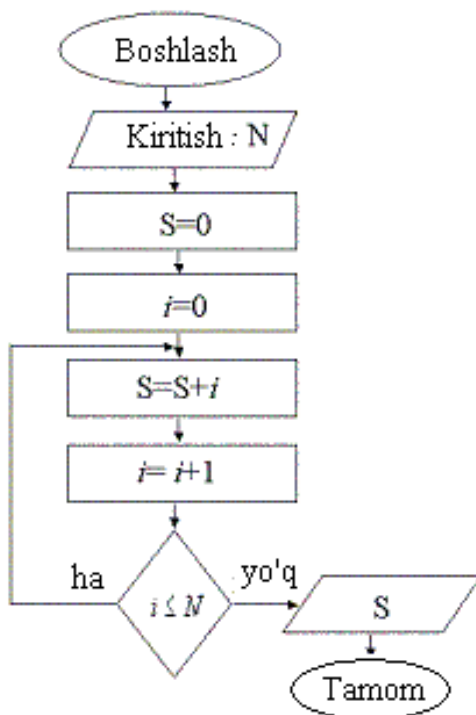
5-rasm. Aylanish strukturasi umumiy ko'rinishi

Takrorlanuvchi algoritmlar. Agar biror masalani yechish uchun tuzilgan zarur bo'lgan amallar ketma-ketligining ma'lum bir qismi biror parametrga 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.

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

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 algoritm sifatida 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.



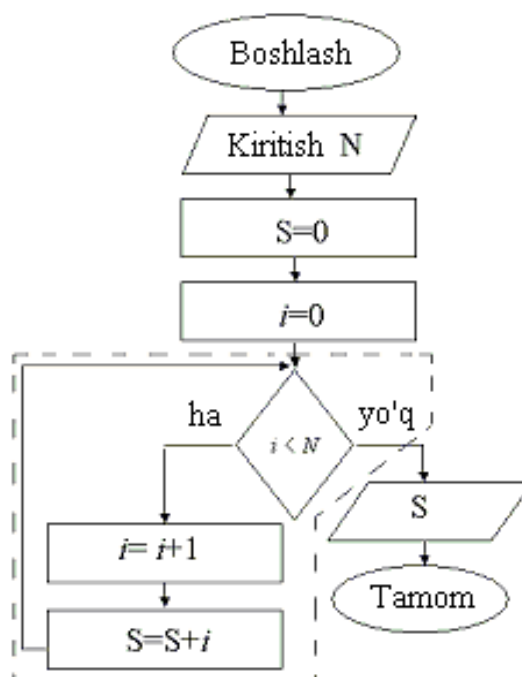
6-rasm. 1 dan n gacha bo'lgan sonlar yig'indisini hisoblash algoritmi

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.

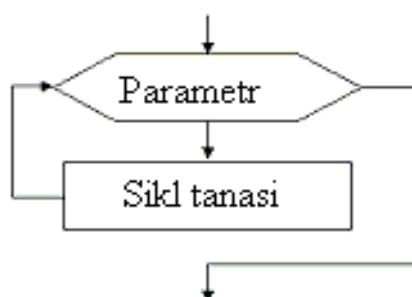
Yuqoridagi blok sxemalarda shartni oldin tekshiriladigan holatda chizish mumkin edi. Masalan, yig'indining algoritmini qaraylik. Bu blok sxemaning

takrorlanuvchi qismiga quyidagi, sharti oldin berilgan siklik strukturaning mos kelishini ko‘rish mumkin.



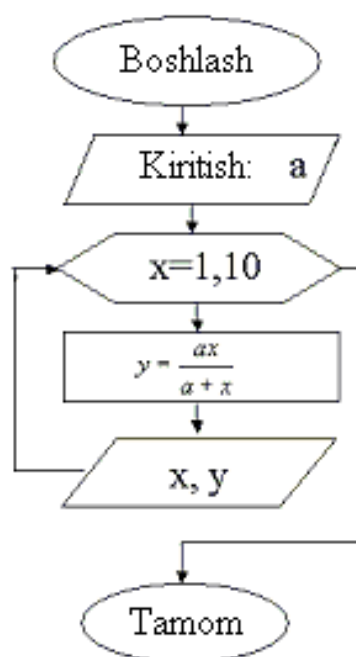
7-rasm. 1 dan n gacha bo‘lgan sonlar yig‘indisini hisoblash algoritmi

Blok sxemalarining takrorlanuvchi qismlarini, quyidagi parametrlilik takrorlash strukturasi ko‘rinishida ham ifodalash mumkin.



8-rasm. Parametrlilik takrorlash operatorining umumiy ko‘rinishi

Parametrlilik takrorlash operatoriga *misol* sifatida berilgan $x=1,2,3,\dots,10$ larda $y = \frac{ax}{a+x}$ funksiyasining qiymatlarini hisoblash blok sxemasini qarash mumkin.



9-rasm. Parametrli takrorlash operatoriga doir algoritm

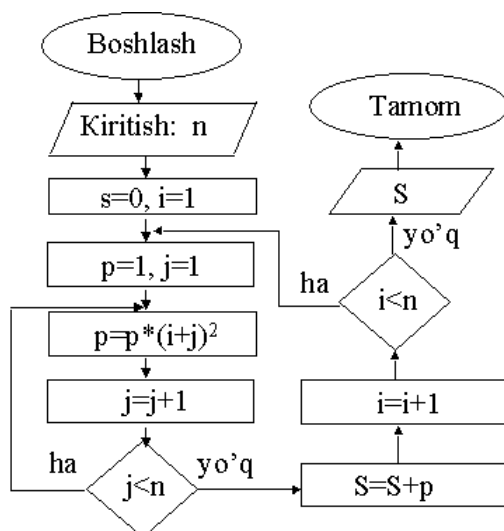
Ichma-ich joylashgan siklik algoritmlar. Ba'zan, takrorlanuvchi algoritmlar bir nechta parametrlarga bog'liq bo'ladi. Odatda bunday algoritmlarni ichma-ich joylashgan algoritmlar deb ataladi.

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

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

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.

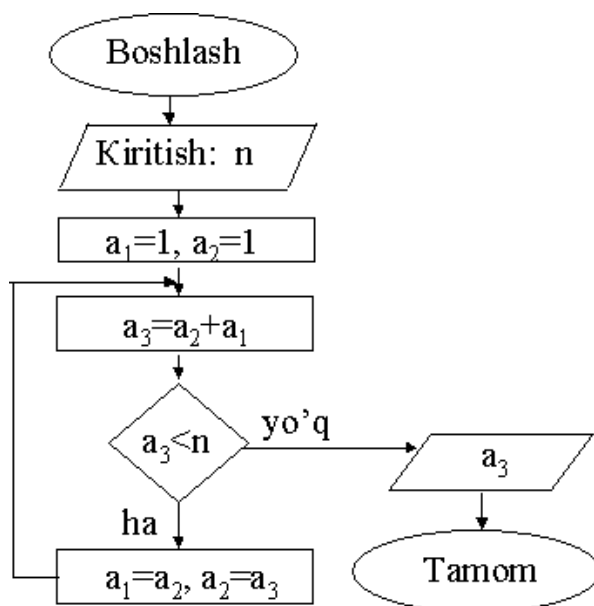


10-rasm. Ichma-ich joylashgan siklik algoritmgaga doir blok-sxema

Rekurent algoritmlar. Hisoblash jarayonida ba'zi bir algoritmlarning o'ziga qayta murojaat qilishga to'g'ri keladi. O'ziga-o'zi murojaat qiladigan algoritmlarga rekurrent algoritmlar yoki rekursiya deb ataladi.

Bunday algoritmgaga *misol* sifatida Fibonachchi sonlarini keltirish mumkin. Ma'lumki, Fibonachchi sonlari quyidagicha aniqlangan.

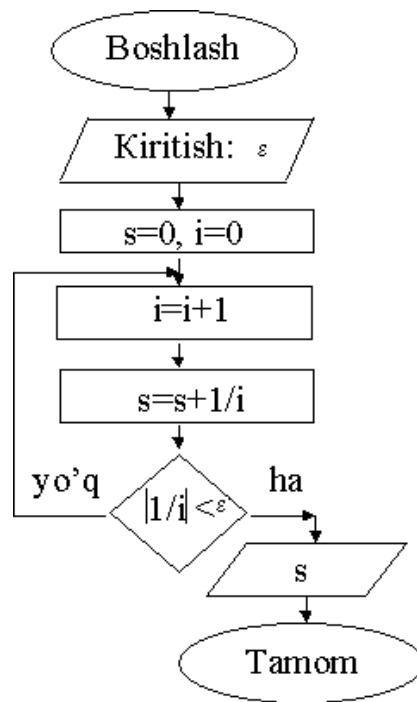
$a_0=0, a_1=1, a_i=a_{i-1}+a_{i-2} \quad i=2,3,4,\dots$ Bu rekurrent ifoda algoritmgaga mos keluvchi blok-sxema 2.15-rasmda keltirilgan. Eslatib o'tamiz formuladagi i -indeksga hojat yo'q, agar Fibonachchi sonining nomerini ham aniqlash zarur bo'lsa, birorta parametr-kalit kiritish kerak bo'ladi.



11-rasm. Fibonachchi sonlarining n - hadini hisoblash algoritmi.

Amalda shunday bir masalalar uchraydiki, ularda takrorlanishlar soni oldindan berilmagan-noma'lum bo'ladi. Ammo, bu jarayonni tugatish uchun biror bir shart berilgan bo'ladi.

Masalan, quyidagi $S = 1 + \frac{1}{2} + \frac{1}{3} + \dots = \sum_{i=1}^{\infty} \frac{1}{i}$ qatorda nechta had bilan chegaralanish berilmagan. Lekin qatorni ε aniqlikda hisoblash zarur bo'ladi. Buning uchun $\left| \frac{1}{i} \right| < \varepsilon$ shartni olish mumkin.



12-rasm. Takrorlanishlar soni oldindan no'malum bo'lgan algoritmlarga doir blok-sxema.

Ketma-ket yaqinlashuvchi yoki iteratsion algoritmlar. Yuqori tartibli algebrayik va transsendent tenglamalarni yechish usullari yoki algoritmlari ketma-ket yaqinlashuvchi – iteratsion algoritmlarga *misollar* bo'la oladi. Ma'lumki, transsendent tenglamalarni yechishning quyidagi asosiy usullari mavjud:

- Urinmalar usuli (Nyuton usuli),
- Ketma-ket yaqinlashishi usuli,
- Vatarlar usuli,
- Teng ikkiga bo'lish usuli.

Bizga

$$f(x)=0 \quad (1)$$

transsendent tenglama berilgan bo'lsin. Faraz qilaylik bu tenglama $[a,b]$ oraliqda uzluksiz va $f(a)*f(b)<0$ shartni qanoatlantirsin. Ma'lumki, bu holda berilgan tenglama $[a,b]$ orilaqda kamida bitta ildizga ega bo'ladi va u quyidagi formula orqali topiladi.

$$X_{n+1} = X_n - \frac{f(X_n)}{f'(X_n)} \quad n = 0,1,2,\dots \quad (2)$$

Boshlang'ich X_0 qiymat $f(x_0)f''(x_0)<0$ shart asosida tanlab olinsa, (2) iteratsion albatta yaqinlashadi. Ketma-ketlik

$$|X_{n+1} - X_n| < \varepsilon$$

shart bajarilgunga davom ettiriladi.

Berilgan musbat a xaqiqiy son dan kvadrat ildiz chiqarish algoritmi tuzilsin.

Bu masalani yechish uchun kvadrat ildizni x deb belgilab olib,

$$\sqrt{a} = x \quad (3)$$

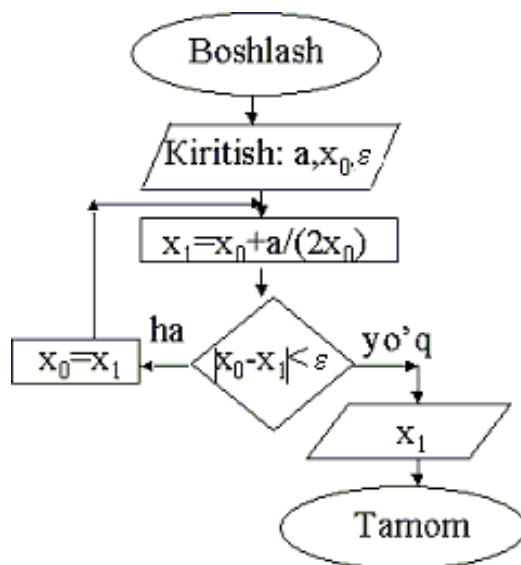
ifodalash yozib olamiz. U holda (1) tenglamaga asosan

$$f(x) = x^2 - a \quad (4)$$

ekanligini topish mumkin (4) ifodani (2) ga qo'yib, quyidagi rekurrent formulani topish mumkin:

$$X_{n+1} = \frac{1}{2} \left(X_n + \frac{a}{2X_n} \right) \quad (5)$$

Bu formulaga mos blok-sxema 2.18-rasmda keltirilgan. ε - kvadrat ildizni topishning berilgan aniqligi. Eslatib o'tamiz, algoritmda indeksli o'zgaruvchilarga zarurat yo'q.

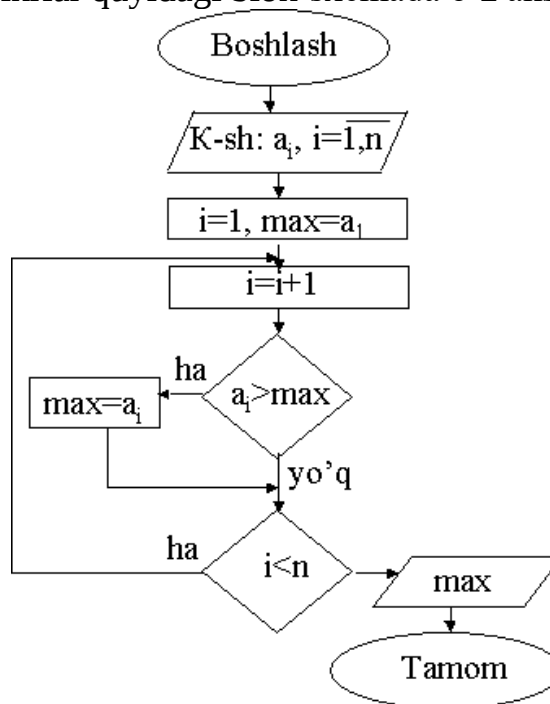


13-rasm. Berilgan musbat a haqiqiy sondan kvadrat ildiz chiqarish algoritmi (iteration algoritmgaga doir blok-sxema).

Algoritm ijrosini tekshirish. Kompyuter uchun tuzilgan algoritm ijrochisi-bu kompyuterdir. Biror programmalash tilida yozilgan algoritm kodlashtirilgan oddiy ko'rsatmalar ketma-ketligiga o'tadi va mashina tomonidan avtomatik ravishda bajariladi. Metodik nuqtayi-nazardan qaraganda algoritmning birinchi ijrochisi sifatida o'quvchining o'zini olish muhim ahamiyatga ega. O'quvchi tomonidan biror masalani yechish algoritmi tuzilganda bu algoritmni to'g'ri natija berishini tekshirish juda muhimdir. Buning yagona usuli o'quvchi tomonidan algoritmni turli boshlang'ich ma'lumotlarda qadamba - qadam bajarib (ijro etib) ko'rishdir. Algoritmni bajarish natijasida xatolar aniqlanadi va to'g'rilanadi. Ikkinchi tomonidan, masalani yechishga qiynalayotgan o'quvchi uchun tayyor algoritmni bajarish – masalani yechish yo'llarini tushunishga xizmat qiladi. Algoritm ijrosini quyidagi *misolda* ko'raylik.

Berilgan a_i , $i=1, n$ sonlarning eng kattasini topish algoritmini tuzaylik. Buning uchun, berilgan sonlardan birinchisi a_1 ni $i=1$ eng katta qiymat deb faraz qilaylik va uni max nomli yangi o'zgaruvchiga uzataylik: $max=a_1$. Parametr i ning qiymatini bittaga oshirib, ya'ni $i=i+1$ a_1 ni a_2 bilan taqqoslaymiz va qaysi biri katta

bo'lsa uni max o'zgaruvchisiga uzatamiz va jarayonni shu tarzda to $i=n$ bo'lguncha davom ettiramiz. Bu fikrlar quyidagi blok-sxemada o'z aksini topgan.



14-rasm. Vektor elementlarining eng kattasini topish algoritmi.

Endi bu blok-sxema yoki algoritmnining ijrosini $n=3$ $a_1=3$, $a_2=5$, $a_3=1$ aniq sonlarda ko'rib o'taylik:

$i=1$ da $max=3$ bo'ladi.

$i=i+1=2$ ni topamiz,

$a_2 > max$, ya'ni $5 > 3$ ni tekshiramiz, shart bajarilsa, $max=5$ bo'ladi.

$i < n$, ya'ni $2 < 3$ ni tekshiramiz. Shart bajarilsa, i ni yana bittaga oshiramiz, va $i=3$ bo'ladi, va

$a_3 > max$, ya'ni $1 > 5$, ni tekshiramiz. Shart bajarilmadi, demak, keyingi

$i < n$ shartni, ya'ni $3 < 3$ ni tekshiramiz. Shart bajarilmadi. Demak $max=5$ chop etiladi. Biz blok-sxemani tahlil qilish davomida uning to'g'riligiga ishonch hosil qildik. Endi ixtiyoriy n lar uchun bu blok-sxema bo'yicha eng katta elementni topish mumkin.

Foydalanilgan adabiyotlar:

1. Informatika va programmalash. O'quv qo'llanma. Mualliflar: A.A.Xaldjigitov, Sh.F.Madraximov, U.E.Adambayev, O'zMU, 2005 yil, 33-52 b.
2. Pascal tilida programmalash bo'yicha masalalar to'plami. O'quv qo'llanma. Mualliflar: A.A.Xaldjigitov, Sh.F.Madraximov, A.M.Ikromov, S.I.Rasulov, O'zMU, 2005 yil, 23-45 b.

3. Amaliy matematikadan kirish lelsiyalari. A.НТихонов,
Д.П.Костомаров. Toshkent. O'qituvchi, 1987. 27-36 b.
4. Вычислительнаа техника и программирование. А.В.Петров.
М.: Просвещение, 1991. 34-58 с.