

O'ZBEKISTON RESPUBLIKASI
OLIY VA O'RTA-MAXSUS TA'LIM VAZIRLIGI

ALISHER NAVOIY NOMIDAGI SAMARQAND DAVLAT
UNIVERSITETI

MEXANIKA-MATEMATIKA FAKULTETI

AMALIY MATEMATIKA VA INFORMATIKA YO'NALISHI

1 KURS 102-GURUH TALABASI

Mamarasulov Rahmiddin

MAXSUS ALGORITMIK TIL ASOSLARI
MAVZUSIDAN

REFERAT

Tayyorladi: Mamarasulov R

Tekshirdi: Nazarov F.

SAMARQAND-2016

MAXSUS ALGORITMIK TIL ASOSLARI. BLOKLASH SXEMASI.

REJA:

1. Algoritm tushunchasi.
2. Algoritm ijrochisi.
3. Algoritmning xossalari.
4. Algoritmni tasvirlash usullari.
5. Algoritm tarkibi.
6. Chiziqli hisoblash jarayonlari.

1 Algoritm tushunchasi.

Algopitm — ijrochi uchun ma'lum bir masalani yechishga qaratilgan ko'rsatmalarning aniq ketma-ketligi.

Algoritm — informatika va matematikaning asosiy tushunchalaridan hisoblanadi.

Algoritm so'zi va tushunchasi algorithmi so'zidan olingan bo'lib, u IX (783 yilda to'g'ilgan) asrda yashab ijod etgan buyuk bobokolonimiz Muxammad al-Xorazmiy nomi bilan uzviy bog'lik bo'lib, uning arifmetikaga bag'ishlangan «Al jabr va al mukabala» nomli asarining dastlabki saxifasidagi «Dixit Algoritmica» («Dediki Ad-Xorazmiy» ning lotincha ifodasi) degan so'zlardan kelib chiqqan.

Al-Xorazmiy birinchi bo'lib unlik sanoq tizimining prinsiplarini va unda turli amallar bajarish qoidalarini asoslab berdi. Bu esa hisoblash ishlarini ixchamlashtirish va osonlashtirish imkonini yaratdi. Chunki bu bilan usha davrda qullanib kelingan Rim raqamlari va sonlarni so'z orkali yozib bajarishdagi noqulayliklar bartaraf etildi.

Algoritm nima? Dastlab algoritm deyilganda unlik sanoq tizimidagi sonlar ustida turli arifmetik amallar bajarish qoidalari tushinib kelingan. Umuman olganda, uni aniq ta'riflash mushkul. Algoritm deganda biror maqsadga erishishga yoki qandaydir masalani yechishga qaratilgan buyruqlarning aniq, tushinarli, chekli hamda to'liq tizimi, aniq natijaga olib keladigan amallarning cheklangan ketma-ketligi tushiniladi.

Algoritimning xizmati nimadan iborat? Aytaylik, kimdir qandaydir masalani yechishni o'ylab topib va uni boshqalarga aytmoqchi bo'lsa, u holda u o'ylab topgan yechimni shunday tasvirlashi kerakki, natijada boshqalar ham masalani to'g'ri yechishsin. Shuning uchun tasvir bir necha talablarga bo'ysunishi kerak. Agar yechimning tasviri aniq bo'lmasa, ya'ni mujmal bo'lsa, u holda shu tasvirga asosan boshqa javobni olish mumkin. Chunki har kim masala yechimning tasvirini noaniq joyini o'zicha aniqlashtirishi mumkin. Bunday tasvirni algoritim deb bo'lmaydi.

Har kuni necha martalab bajaradigan ishimiz algoritmgga misol bo'la oladi. Algoritmni ishlab chiqish uchun avvalo masalaning yechish yo'lini yaxshi tasavvur qilib olish, keyin esa uni formallashtirish, ya'ni aniq qoidalar ketma-ketligi ko'rinishida yozish kerak.

Bu misollardan bitta umumiy tomonni kuzatishimiz mumkin. Bu algoritmdan qanday maqsad ko'zlanganligini bilmasdan turib ham uni muvaffaqiyat bilan bajarish mumkin. Demak, hayotda uchraydigan murakkab jarayonlarni boshqarishni yoki amalga oshirishni robotlar, kompyuterlar va boshqa mashinalar zimmasiga yuklashimiz mumkin ekan. Bu algoritmning juda muhim afzalligidir. Shunga kura, har bir inson o'z oldiga quyilgan masalaning yechish algoritmini to'g'ri tuzib bera olsa, u o'z aqliy va jismoniy mehnatini yengillashtiribgina qolmay, bu ishlarni avtomatik tarzda bajarishni mashinalarga topshirishi mumkin.

Svetofordan foydalanish algoritmi.

- 1) svetofor chirog'iga qaralsin;
- 2) qizil chiroq yongan bo'lsa tuxtalsin;
- 3) sariq chiroq yongan bo'lsa, yurishga yoki tuxtashga tayyorlansin;
- 4) yashil chiroq yongan bo'lsa yurilsin.

2. Algoritm ijrochisi.

Algoritm ijrochisi — algoritmda ko'rsatilgan buyruqlarni bajara oladigan abstrakt yoki real (texnik, biologik yoki biotexnik) tizim.

Algoritmarga xos xususiyatlar:

oddiy xarakatlar;

buyruqlar tizimi.

✓ **Buyruqlar tizimi.** Har bir ijrochi faqatgina ushbu ijrochi tushunadigan buyruqlarni (ya'ni, ijrochi bajaradigan buyruqlar ro'yxatiga mansublarni) bajara oladi.

✓ Ijrochi buyruqlarni bajarish jarayonida **oddiy harakatlarni bajaradi.**

Odatda ijrochiga algoritmning maqsadi ma'lum bo'lmaydi. Shuning uchun

ijrochi “nimaga” va “nima uchun” degan savollarni bermaydi.

Informatikada algoritmlarning universal ijrochisi, **kompyuterlar** hisoblanadi.

3. Algoritmning xossalari.

Algoritmlarning asosiy hossalari quyidagilardan iborat:

❶ **Tushunarlilik.** Algoritm ijrochisi buyruklar ketma-ketligini qanday bajarishni aniq bilishi kerak. Algoritmning ijrochisi hamma vaqt ham inson bo'lavermaydi. Ijrochiga tavsiya etilayotgan ko'rsatmalar uning uchun tushinarli bo'lishi kerak, aks holda ijrochi har qanday amalni ham bajara olmaydi.

❷ **Diskpyetlik.** Algoritm ijrochisi masalani yechish jarayonini alohida va sodda qadamlar ketma-ketligini bajarish deb tushunishi kerak. Bu xossaning mazmuni – algoritmlarni doimo chekli qadamlardan iborat qilib bulaklash imkoniyati mavjudligidadir. Boshqacha aytganda, uni chekli sondagi oddiy ko'rsatmalar ketma-ketligi shaklida ifodalash mumkin. Algoritmning bu hossasi yuqorida keltirilgan misollarda yaqqol ko'rinib turibdi. Agar ko'zatilayotgan jarayonlarni chekli qadamlardan iborat qilib bulaklay olmasak, u holda uni algoritm deb bulmaydi.

❸ **Aniqlik.** Algoritmning xar bir qoidasi, undagi amallar va buyruqlar bir ma'noli bo'lishi kerak. Shu hossaga asosan algoritm ijrochisi buyruqlar ketma-ketligini mexanik bajarish imkoniyatiga ega bo'ladi. Ijrochiga berilayotgan ko'rsatmalar aniq mazmunda bulishi kerak. Chunki, ko'rsatmadagi noaniqliklar muljaldagi maqsadga erishishga olib kelmaydi.

❹ **Natijaviylik.** Bu hossaning mazmuni shundan iboratki, har qanday algoritmning ijrosi oxir-oqibat ma'lum bir yechimga kelishi kerak. Har bir algoritm chekli sondagi qadamlardan keyin, albatta, natija berishi shart. Bajariladigan amallar ko'p bo'lsa ham, baribir natijaga olib kelishi kerak. Chekli qadamdan keyin qo'yilgan masala yechimga ega emasligini aniqlash ham natija hisoblanadi. Agar ko'rilyotgan jarayon cheksiz davom etib, natija bermasa, uni algoritm deb ayta olmaymiz.

❺ **Ommaviylik.** Masalani yechish algoritmi umumiy hollar uchun yaratiladi, ya'ni faqatgina boshlang'ich qiymatlari bilan farqlanuvchi bir turdagi masalalar sinfi uchun tuziladi. Bunda boshlang'ich qiymatlar algoritmning qiymatlar qabul qilishi mumkin bo'lgan sohadan olinadi. Har bir algoritm mazmuniga

ko'ra bir turdagi masalalarning barchasi uchun ham urinli bo'lishi kerak, ya'ni masalaning boshlang'ich ma'lumotlari qanday bo'lishidan qat'iy nazar algoritmlar shu xildagi har qanday masalani yechishga yaroqlidir.

4. Algoritmni tasvirlash usullari.

Amaliyotda algoritmlarni tasvirlashning keng tarqalgan usullari quyidagilar:

1. **So'zlar yordamida** (og'zaki nutqda ishlatiladigan so'zlar) Algoritmning so'z orqali berilishi. Bunda ijrochi uchun beriladigan har bir ko'rsatma so'zlar orqali buyruq mazmunida beriladi;

2. **Algoritmning formulalar yordamida berilishi.** Algoritmning formulalar bilan berilish usulidan matematika, fizika, kimyo va boshqa aniq fanlarni o'rganishda keng foydalaniladi. Masalan, uchburchakning yuzini uning asosi va balandligi bo'yicha hisoblash formulasi: $S = (h \cdot a) / 2$

Algoritmning jadval ko'rinishida berilishi. Algoritmni bu ko'rinishida tasvirlanishidan ham keng foydalaniladi.

3. **Grafik usulda** (grafik simvollar yordamida) Algoritmning grafik (blok-sxema) shaklida tasvirlanishi. Algoritmning blok-sxema ko'rinishdagi tasvirida geometrik figuralar shaklidagi oddiy elementlardan foydalaniladi.

4. **Dastur ko'rinishida** (dasturlash tillariga oid xizmatchi so'zlar, operator va funksiyalar yordamida). Algoritmning dastur shaklida ifodalanishi. Millionlab kompyuterlarning keng tarqalib ketishi algoritmning dastur tarzida tasvirning keng ommalashib ketishiga katta turtki berdi. Sababi shundaki, kompyuterlar doimo dastur orqali boshqariladi.

Algoritmni so'zlar yordamida tasvirlash.

Algoritmni so'zlar yordamida tasvirlashda bajariladigan buyruqlar va ko'rsatmalar ketma-ket og'zaki nutqda ishlatiladigan so'zlar orqali yoziladi.

Masalan, ikki sonning **eng katta umumiy buluvchisini (EKUB) topish** algoritmi quyidagicha yozilishi mumkin:

Ikkita sonni kiriting;

Agarda bu sonlar teng bo'lsa, u holda ulardan birini javob sifatida oling va ishini tuxtatib, aks holda esa davom ettiring;

Ikkitasini ichida kattasini aniqlang;

Katta va kichik sonlarning ayirmasini katta son bilan almashtiring;

Algoritmni 2-qadamdan boshlab qaytaring.

Keltirilgan algoritmnı har qanday natural sonlarning EKUBini topish uchun ishlatish mumkin.

Algoritmnlarnı soʻzlar yordamida tasvirlashning bir qancha kamchiligi mavjud boʻlib, aksariyat hollarda algoritmnlarnı tasvirlashda bu usuldan foydalanilmaydi.

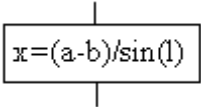
Algoritmnlarnı grafik usulda tasvirlash.

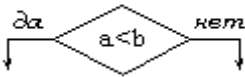
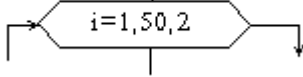
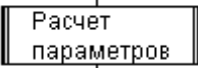
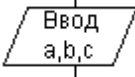
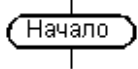
Algoritmnlarnı grafik usulda tasvirlashda har bir amal bir yoki bir nechta harakatni ifodalovchi oʻzaro bogʻliq funksional bloklar ketma-ketligi orqali tasvirlanadi.

Algoritmning bunday tasvirlash usuli algoritm sxemasi yoki **blok-sxema** deb ataladi.

Blok-sxemada har bir xarakat turini (boshlangʻich qiymatlarnı kiritish, ifodalar qiymatlarini hisoblash, shartlarnı tekshirish, amallarnı takrorlashni boshqarish, qayta ishlashni tugatish va x.k.) maʼlum bir geometrik figura orqali ifodalanadi.

Blokli belgilar (geometrik figuralar) chiziqlar orqali bogʻlanadi (bunda qaysi amal oldin, qaysinisi keyin bajarilishi kursatiladi). Nisbatan murakkab masalalarnı yechishda algoritmdan muayan EHM tilidagi dasturga utish juda qiyin. Bunday bevosita utishda algoritmning alohida qismlari orasidagi bogʻlanish yoʻqoladi, algoritm tarkibining asosiy va muhim boʻlmagan qismlarini farqlash qiyin boʻlib qoladi. Bunday sharoitda keyinchalik aniklash va toʻgʻrilash ancha vaqt talab qiladigan xatolarga osongina yoʻl qoʻyish mumkin. Odatda, algoritm bir necha marta ishlab chiqiladi, baʼzan xatolarnı toʻgʻrilash, algoritm tarkibini aniqlashtirish va tekshirish uchun bir necha marta orqaga qaytishga toʻgʻri keladi. Algoritm ishlab chiqishning birinchi bosqichida algoritmni yozishning eng qulay usuli algoritmni blok-sxema koʻrinishda ifodalashdir.

1 jadval blok-sxemada ishlatiladigan bloklarnı aks etadi. Amallarnı belgilanishi.	Izoh
	Oddiy harakat

	Shart tekshirish
 i=1,50,2	Sikl (takrorlanish) boshi
	Yordamchi algoritmgga murojaat
	Ma'lumotlarni kiritish va chiqarishning umumiy ko'rinishi
	Algoritmning boshi va oxiri
	Natijani bosmaga chiqarish

♥"Oddiy xarakat" belgisi orqali formulalar, hisob-kitob, o'zlashtirish amallari ifodalaniladi. Bir nechta amallarni alohida yoki bitta belgi orqali ifodalash mumkin.

♥"Shart tekshirish" bloki orqali amallar bajarilish yo'nalishi shart bajarilishi asosida ko'rsatiladi. Bunday blokning har birida savol, shart yoki munosabat ko'rsatiladi.

♥"Sikl" bloki amallarni takrorlash uchun ishlatiladi. Blok ichida siklning boshi va oxirini ko'rsatuvchi parametr (**i**), parametrning o'zgarish qadami ko'rsatiladi.

♥"Yordamchi algoritmgga murojaat" bloki alohida va mustaqil ishlovchi qism dastur va yordamchi algoritmlarga murojaatni bildiradi.

5. Algoritm tarkibi.

Algoritm tuzilishi

Har qanday algoritmning mantiqiy tuzilishi uchta asosiy elementlar orqali ifoda qilinishi mumkin:

ketma-ketlik, tarmoqlanish, takrorlanish (sikl).

EXMlarning taraqqiyoti uchun algoritm va dasturlashning tutgan urni.

EXMlarning rivojlanishi bilan dasturlarni avtomatlashtirish, ularni dasturlarni yozishda qulay qilib ishlab chiqish asosiy mutaxassislarning asosiy masalalaridan biri bo'lib qoldi.

Bu yo'nalishlardan birinchi qadamlardan biri mashinaga-muljallangan tillardan biri bulgan Assemblerni ishlab chiqdi.

Dasturlar ustida keyingi ishlab chiqishlar mashinaga bog'liq bo'lmagan tillar hisoblanib, bular orqali ixtiyoriy EHMLarda dastur tuzish imkonini yaratdi. Bu esa muhandislik, ilmiy texnik, iqtisodiy axborotlarni qayta ishlovchi, ro'yxatlarni qayta ishlovchi modellashtirishda ishlatiluvchi va xokozo prosedurali – yo'naltirilgan tillarni yaratilishiga sabab bo'ldi. Bu tillar qatoriga Paskal, Fortran, Algol, Kobol va shu kabi tillar kiradi.

Xozirgi kunda hisoblash, muhandis – texnik, iqtisodiy, matnli va sonli axborotlarni tahlil qilish va boshqa masalalarni yechish uchun yuqori darajadagi dasturlash tillari mavjud. Bular jumlasiga Beysik, Fortran, SI, S++ va boshqalar kiradi.

Xozirda dasturlash tillarini quyidagi turlarga ajratishimiz mumkin:

- mashinaga –muljallangan Assembler va boshq.
- proseduraga muljallangan Fortran, Kobol va boshq.
- muammoga muljallangan PL/1 va boshqalar.
- dialogli Beysik va boshq.
- strukturali dasturlovchi Paskal va boshq.
- mantiqiy dasturlovchi Prolog va boshq.

FOYDALANILGAN ADABIYOTLAR

1. O`.T.Haitmatov va b. Informatika va axborot texnologiyalari. O`quv qo`llanma. T. TKTI. 2005 y.
2. O`.T.Haitmatov va b. Informatika va axborot texnologiyalari fanidan laboratoriya ishlarini bajarish ushuni uslubiy qo`llanma. T. TKTI. 2005 y.
3. Faronov V.V. Turbo Paskal 7.0. Uchebnoe posobie. M.: Nolidj., 2002g.
4. Aripov M., Xaydarov A. Informatika asoslari T. "O`qituvchi" 2002y.
5. Holmatov T.X., Toyloqov N.I. Amaliy matematika, dasturlash va kompyuterning dasturiy ta'minoti. T. Mexnat, 2000 y.