

A.R. Azamatov

A36 об.аз.

408209



ALGORITMLASH VA DASTURLASH ASOSLARI



O'ZBEKISTON RESPUBLIKASI OLIY VA O'RTA
MAXSUS TA'LIM VAZIRLIGI
O'RTA MAXSUS, KASB-HUNAR TA'LIMI MARKAZI

A.R. AZAMATOV

ALGORITMLASH VA DASTURLASH ASOSLARI

Kasb-hunar kollejlari uchun o'quv qo'llanma

Ikkinchi nashri

*Cho'lpon nomidagi nashriyot-matbaa ijodiy uyi
Toshkent — 2011*

UDK: 519.711(075)

BBK 22.12ya722

A36

*Oliy va o'rta maxsus kasb-hunar ta'limi o'quv metodik
birlashmalar faoliyatini muvofiqlashtiruvchi Kengash
nashrga tavsiya etgan*

Taqrizchi:

B. Boltayev — fizika-matematika fanlari nomzodi.

Azamatov, A. R.

A36

Algoritmlash va dasturlash asoslari: kasb-hunar kollejlari uchun o'quv qo'llanma/A.R. Azamatov; O'zbekiston Respublikasi Oliy va o'rta maxsus ta'lim vazirligi, O'rta maxsus, kasb-hunar ta'limi markazi. Ikkinchi nashri. — T.: Cho'lpon nomidagi nashriyot-matbaa ijodiy uyi, 2011. — 232 b.

ISBN 978-9943-05-439-4

O'quv qo'llanma «Algoritmlash va dasturlash asoslari» o'quv fani dasturi asosida yozildi. Unda algoritmlash asoslarini o'rgatish uchun Ijrochilar va ularning ko'rsatmalar sistemasi kiritilgan bo'lib, algoritmlar tushunchasi, algoritmlar xossalari, ularni tasvirlash va tuzish usullari haqida ma'lumotlar yoritilgan. Bundan tashqari, ko'rsatmalarni dasturlash tillarida ifodalari izohlanib, BASIC, PASCAL va DELPHI dasturlash tillarida taqqoslamalar berilgan. Nazariy ma'lumotlardan tashqari takrorlash va mustaqil ishlash uchun ko'p sonli vazifalar keltirilgan.

Ushbu o'quv qo'llanma kasb-hunar kollejlari va akademik litsey talabalari uchun mo'ljallangan bo'lib, undan algoritmlash va dasturlashni o'rganmoqchi bo'lgan barcha kitobxonlar ham foydalanishlari mumkin.

UDK: 519.711(075)

BBK 22.12ya722

ISBN 978-9943-05-439-4

© Cho'lpon nomidagi nashriyot-matbaa ijodiy uyi, 2010

© Cho'lpon nomidagi nashriyot-matbaa ijodiy uyi, 2011

I bob. ALGORITMLAR

Algoritm tushunchasi

Inson hayoti davomida katta-kichik vazifalar yoki masalalarni hal etishni o'z oldiga maqsad qilib qo'yadi. Odatda, u o'z maqsadiga erishishi uchun bajarishi lozim bo'lgan amal yoki ishlarini hayotiy tajribasi yoki o'zlashtirgan bilimiga asoslanib ma'lum bir tartibga keltiradi. Bunga hayotimizdan xilma-xil misollar keltirish mumkin.

1.1-misol

Ko'chadan o'tish maqsad qilib qo'yilgan bo'lsin. U holda ko'chadan o'tayotgan kishi hammamizga odatiy hol bo'lib qolgan quyidagi harakatlarni bajarishi lozim bo'ladi:

- 1) chap tarafga qaralsin, agar transport vositasi yo'q bo'lsa, 2-bandga o'tilsin, aks holda 1-bandga o'tilsin;
- 2) o'ng tarafga qaralsin, agar transport vositasi yo'q bo'lsa, 3-bandga o'tilsin, aks holda 1-bandga o'tilsin;
- 3) ko'chadan o'tilsin.

1.2-misol

Eni 6 metr va bo'yi 10 metr bo'lgan joyni to'ldirish uchun sotib olinishi kerak bo'lgan 12x25 sm (eni 12 sm va bo'yi 25 sm) g'ishtlar soni topilsin.

Hisoblayotgan kishi geometriya fanidan olgan bilimiga asoslanib quyidagi ketma-ketlikdagi amallarni bajaradi:

- 1) joyning yuzasi S_{joy} santimetr o'lchov birligida topilsin;
- 2) bir dona g'ishtning yuzasi $S_{g'isht}$ santimetr o'lchov birligida topilsin;
- 3) g'ishtlar soni S_{son} joyning yuzasini g'ishtning yuzasiga nisbati deb olinsin.

Bu amallar ketma-ketligini quyidagi matematik formula bilan ham ifodalash mumkin:

$$S_{son} = \frac{S_{joy}}{S_{g'isht}} = \frac{6 \cdot 100 \cdot 10 \cdot 100}{12 \cdot 25}.$$

1.3-misol

Amal bajarisin: 19632107 + 19702202. Bu amalni qanday bajargan bo'lar edingiz? Ha, to'g'ri, bu sonlarni ustun ko'rinishida deyarli quyidagicha qo'shasiz:

1) sonlar xonalari to'g'ri keladigan tartibda birinchisining tagiga ikkinchisi yozib olinsin;

2) sonlarning birlik xonasidagi raqamlarini qo'shib, natijani birlik xonasidagi raqami birliklar tagiga yozilib, o'nlik xonasi raqami dilda saqlansin;

3) sonlarning o'nliklardagi va dildagi raqamlarni qo'shib, natijani birlikdagi raqami o'nliklar tagiga yozilib, o'nlik raqami dilda saqlansin;

va 3-banddagi qoida yuzliklar, mingliklar va hokazo uchun takrorlanadi. Bu amallar quyidagi korinishda sizga juda tanish:

$$\begin{array}{r} 19632107 \\ + 19702202 \\ \hline 39334309 \end{array}$$

Yuqoridagi misollarda keltirilgan amallar ketma-ketligi, boshqacha aytganda, ko'rsatmalar yoki buyruqlar ketma-ketligi biror kishi tomonidan bajarilgach, ko'zlangan maqsadga erishiladi. Bunday amallar ketma-ketligi yoki hayotimizda har kuni va har soatda uchrab turadigan turli qoidalar ichida biror zaruriy natijaga erishishga olib keladigan amallarni ketma-ket bajarishni talab etadigan qoidalar informatikaning asosiy tushunchalaridan biri **algoritm** so'zi bilan ifodalanadi.

Algoritm so'zi IX asrda yashab (783- yilda tug'ilgan) o'z ilmiy ishlari xazinasi bilan dunyoga tanilgan vatandoshimiz buyuk astronom, matematik va geograf Abu Abdullo Muhammad ibn Muso al-Xorazmiy nomidan kelib chiqqan. Al-Xorazmiy arifmetikaga bag'ishlangan «Hind hisobi haqida kitob» risolasida to'q-qizta hind raqamining sonlarni ifodalashdagi afzalliklari va ular yordamida har qanday sonni ham qisqa va oson yozish mumkinligini aytadi va hozirgi kunda hamma o'quvchilar biladigan sonlar ustida, yuqoridagi 3-misoldagi kabi ustun ko'rinishida amallar bajarish qoidalarini yoritadi. Ayniqsa, nol (0) qo'llashning ahamiyati haqida tushuncha berib, nolni yozmaslik natijaning xato chiqishiga olib keladi, degan. Bu risola XII asrda Ispaniyada lotin tiliga tarjima qilingan va butun Yevropaga tarqatilgan. Bu tarjimaning XIV asrda ko'chirilgan qo'lyozmasining yagona nusxasi Kembrij universitetining kutubxonasida saqlanmoqda. Risola

«*Dixit Alxhorithmi*», ya'ni «*Dediki al-Xorazmiy*» iborasi bilan boshlanadi.



Algoritm deganda, biror maqsadga erishishga qaratilgan iirochi bajarishi uchun mo'ljallangan ko'rsatma (bu'ruq)larning aniq, tushunarli va chekli ketma-ketligi tushuniladi.

Bu algoritm tushunchasining matematik ta'rifi bo'lmasa ham intuitiv ma'noda algoritmning mazmunini ochib beruvchi tavsifidir. Algoritmni intuitiv ma'noda bir necha misollarda izohlaymiz. Biror-bir narsani taqiqlovchi qoidalar algoritm bo'lolmaydi, masalan: «Chekish mumkin emas», «Begonalarning kirishi taqiqlanadi», «Kirish», «Chekish uchun joy» kabi biror-bir narsaga ruxsat etuvchi qoidalar ham algoritmgaga xos emas. Lekin «Svetofozni yashil rangida o'ting» juda sodda bo'lsa ham algoritmdir.

Demak, yuqorida keltirilgan misollardagi ko'rsatmalar ketma-ketligi **algoritm** va bu algoritmlarni bajarayotgan inson — **ijrochi** bo'lar ekan. Algoritm ijrochisi faqat insonmi, degan savol berishingiz tabiiy. Bu savolga javob quyidagicha:



Algoritm ijrochisi — algoritmga ko'rsatilgan bu'ruq voki ko'rsatmalarni bajara oladigan abstrakt voki real (texnik voki biologik) sistema.

Ijrochi bajara olishi uchun algoritm unga tushunarli bo'lishi lozim. Algoritm ijrochi tushunadigan tilgagina emas, balki uning bilim va malakasiga ham mos bo'lishi kerak. Aks holda ijrochi birorta ham ko'rsatmani bajara olmasligi mumkin.

Ijrochi bajara olishi mumkin bo'lgan ko'rsatma yoki buyruqlar to'plami **ijrochining ko'rsatmalar sistemasini** deyiladi. Masalan, «16 sonidan kvadrat ildiz chiqarilsin» ko'rsatmasi 2-sinf o'quvchisining ko'rsatmalar sistemasiga tegishli bo'lmaydi, lekin 8-sinf o'quvchisining ko'rsatmalar sistemasiga tegishli bo'ladi. Algoritm ijrochiga tushunarli bo'lishi uchun ijrochining imkoniyatlarini bilish lozim. Agar ijrochi inson bo'lsa, u holda algoritm insonning imkoniyatlaridan kelib chiqib tuzilishi kerak. Bunda ko'zlangan maqsad va algoritmdan kelib chiqib inson tushunadigan til, insonning bilimi, hayotiy tajribasi, kasbiy malakasi, yoshi, qolaversa, jismoniy imkoniyatlari hisobga olinishi zarur. Agar ijrochi texnik vosita (masalan, kompyuter, elektron soat, dastgohlar) bo'lsa, u holda algoritm shu texnik vositaning imkoniyatlaridan kelib chiqib tuzilishi kerak.

Agar ijrochi kompyuter hisoblanib, uning dasturiy ta'minotida berilgan («Karra jadvalini hosil qilish») algoritmni bajara oladigan

dastur (masalan, elektron jadvallardan birortasi ham) bo'lmasa, u holda hech qanday natijaga erishib bo'lmaydi.

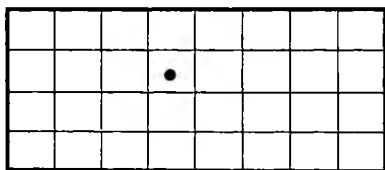
Demak, berilayotgan har qanday ko'rsatma ijrochining ko'rsatmalar sistemasidan olinishi, ya'ni ijrochi uni qanday bajarishni bilishi kerak ekan. Bu algoritmnining **tushunarlilik** xossasini ifodalaydi.

Shuni ta'kidlash joizki, informatikada algoritmnining asosiy ijrochisi sifatida kompyuter xizmat qiladi.

Ijrochi

Bu qo'llanmada algoritm tuzish usullarini o'rgatish uchun sizni bir nechta Ijrochi bilan tanishtiramiz [1], lekin ular kompyuter yoki inson emas, balki biz uchun abstrakt sistema. Misol sifatida **Robot** nomli ijrochi bilan tanishtiramiz.

Robot teng o'lchamdagi kvadratlarga bo'lingan tekislikda yashaydi (1.1-rasm). U kvadratlarning birida joylashgan va ixtiyoriy qo'shni kvadratga o'tishi mumkin. Shu bilan birga Robot o'zi turgan kvadratni bo'yashi mumkin.



1.1-rasm.

Robot 5 ta ko'rsatmani bajaradi:

yuqoriga

quyiga

chapga

o'ngga

bo'ya

Bulardan **yuqoriga**, **quyiga**, **chapga** va **o'ngga** ko'rsatmalari Robotni mos yo'nalishlar bilan siljishga majbur qiladi. Lekin **bo'ya** ko'rsatmasida Robot harakatlanmaydi, faqat o'zi turgan kvadratni bo'yaydi. Agar kvadrat bo'yalgan bo'lsa u holda **bo'ya** ko'rsatmasida kvadratning rangi o'zgarmaydi.

Robotning hayotidagi voqealardan eng qizig'i, ba'zi kvadratlarda devor borligi (1.2-rasm). Odatda, Robot har tomondan devorlar bilan o'ralgan va kvadratlardan hosil bo'lgan to'g'ri

to'rtburchak ichida joylashgan bo'ladi. Lekin shu to'g'ri to'rtburchak ichida ham devorlar bo'lishi mumkin.

Ba'zan devorlar murakkab shaklni hosil qiladi, bu shaklni **labirint** deb atashadi.

Robot devor ichidan o'ta olmaydi. Agar devor ichidan o'tmoqchi bo'lsa, u holda Robot «sochilib» ketadi.



1.2-rasm.

Bunday halokatli holatlarga tushmaslik uchun quyidagi to'rtta shartni tekshirish zarur:

yuqori bo'sh

quyi bo'sh

chap bo'sh

o'ng bo'sh

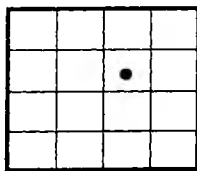
Bo'sh so'zi shu tomonda devor yo'qligini bildiradi.

Robot o'zi turgan katakning devorinigina aniqlay oladi. O'zi turgan kvadrat bilan devor orasida bitta kvadrat bo'lsa ham uzoqdagi bu devorni «ko'ra» olmaydi. U yonida turgan devorgagina «tegib» ko'rish mumkin. 1.3-rasmda turli holatlarda yuqori bo'sh degan birgina shartning qiymatini ko'rish mumkin. Tushunarliki, yuqori bo'sh sharti (yoki yuqori bo'sh da'vosi Rost bo'lsa) Robot yuqoriga ko'rsatmasini «sochilib» ketmasdan bajara olishini bildiradi.

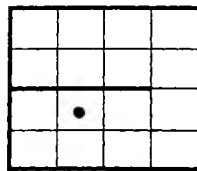
yuqori bo'sh



YOLG'ON



ROST



YOLG'ON

1.3-rasm.

Bu kabi mulohazalar chap bo'sh sharti va chapga ko'rsatmasi, yana boshqa juftliklar uchun ham to'g'ri.

Ro'yxatni yakunlash uchun Robot biladigan oxirgi shartni keltiramiz:

bo'yalgan

Bu shart Robot turgan kvadratni bo'yalgan yoki bo'yal-maganligini tekshirish imkonini beradi. Agar kvadrat bo'yalgan bo'lsa, shart ROST, aks holda YOLG'ON bo'ladi.

Ko'rib turibsiz, Robotning ko'rsatmalari juda sodda. Lekin uni o'rab turgan muhit xilma-xil imkoniyatlarga boy. Robotning maydonida turli labirintlar, yo'laklar, har xil shakldagi xonalar va boshqa figuralar yordamida juda ko'p qiziqarli masalalar qo'ysih mumkin. Robotning mikrohayoti — algoritmik tafakkurni rivojlan-tirish uchun a'lo darajadagi mashq maydonidir.

Ijrochilarni boshqalari bilan tanishtirishdan avval ularni nimalar farqlab turishini izohlab o'tmoqchimiz.

Ijrochini quyidagilar farqlab turadi:

- ijrochi muhiti;
- ijrochining ko'rsatmalar sistemasi;
- sodda amallar;
- INKOR.

Ijrochi muhiti — ijrochi «yashaydigan» yoki algoritmni baja-radigan muhiti. Ijrochi Robot misolida bu katakli maydon, bo'yalgan kataklar va devorlar. Ularning joylashishi va Robotning turgan joyi muhitning aniq holatini beradi.

Har bir ijrochi qat'iy belgilangan ro'yxatdagi — ijrochinining ko'rsatmalar sistemasidagi — ko'rsatmalarni bajara oladi. Har bir ko'rsatma uchun qo'llash sharti (muhitning qanday holatida ko'rsatmani bajarish mumkinligi) va ko'rsatmani bajarilish natijasi belgilangan bo'lishi kerak. Masalan, **yuqoriga** ko'rsatmasi Ro-botning yuqorisida devor yo'q bo'lsagina bajarish mumkin. Bu ko'rsatmani bajarilish natijasi — Robot yuqoriga bitta katak siljiydi.

Ko'rsatma chaqirilgandan keyin Ijrochi **sodda amal** bajaradi. Robot misolida — yuqoriga bitta katak siljish.

INKOR — bu holat bo'lib, ko'rsatma muhitning mumkin bo'lmagan holatida chaqirilganda yuz beradi. Robot misolida qarasak, agar u devor ichidan o'tmoqchi bo'lsa, «sochilib» ketadi va bu Robot uchun INKOR holatiga olib keladi.

Yodingizda bo'lsin: Ijrochi algoritm maqsadi haqida hech narsa bilmaydi, u berilgan ko'rsatmalarni so'zsiz bajaradi, xolos.

Algoritmning xossalari

Algoritmning tavsifida «biror maqsadga erishishga qaratilgan» jumlasini qo'llanilgan. Bu maqsadni yuqorida keltirilgan misollarda ko'rishimiz mumkin: ko'chadan o'tish, g'ishtlar sonini hisoblash, yig'indini hisoblash. Bular algoritmning natijaviylik (cheklilik) xossasi bilan bog'liq. Bu xossaning mazmuni shundan iboratki, har qanday algoritm ijrochi chekli qadamdan so'ng oxir-oqibat ma'lum bir yechimga olib kelishi kerak. Shuni ta'kidlash joizki, algoritm avvaldan ko'zlangan maqsadga erishishga olib kelmasligi ham mumkin. Bunga ba'zan algoritmning noto'g'ri tuzilgani yoki boshqa xatolik sabab bo'lishi mumkin. Ikkinchi tomondan, qo'yilgan masala ijobiy yechimga ega bo'lmasligi ham mumkin. Lekin salbiy natija ham natija deb qabul qilinadi.

1.4-misol

$x^2+x+1=0$ kvadrat tenglama yechilsin.

Bu tenglamaga quyida keltirilgan « $ax^2+bx+c=0$ ($a \neq 0$) ko'rinishidagi kvadrat tenglamani yechish» algoritmini qo'llab, tenglama yechimga ega emasligini aniqlaymiz. Bu ham *natija* ekanligi sizga ma'lum.

1) diskriminant: $D = b^2 - 4ac$ hisoblansin;

2) agar $D < 0$ bo'lsa, tenglama yechimga ega emas deb olinsin va 5-bandga o'tilsin;

3) agar $D = 0$ bo'lsa, yagona yechim $-\frac{b}{2a}$ ga teng deb olinsin va 5-bandga o'tilsin;

4) birinchi yechim $\frac{-b - \sqrt{D}}{2a}$ ga, ikkinchi yechim $\frac{-b + \sqrt{D}}{2a}$ ga teng deb olinsin;

5) tugallansin.

1.1-masha

E'tibor qilingan bo'lsangiz, diskriminantning noldan kichikligi, nolga tengligi tekshirildi, ammo noldan kattaligi tekshirilmadi. Sababini o'zlab ko'ring!

Demak, algoritm doimo chekli qadamdan iborat bo'lishi va biror natija berishi kerak ekan. Bu algoritmnii **diskretlilik** (uzluksizlik,

alohidalik) xossasiga olib keladi. Algoritmida masalani yechish jarayoni alohida olingan sodda ko'rsatmalar ketma-ketligini qadamba-qadam bajarishdan iborat bo'lishi kerak. Bu xossa misollardan yaqqol ko'rinishga ega. «Angliyada avtomobilni yo'lning chap qismida haydang» qoidasi talabnoma bo'lgani bilan uzluksizlik xarakteriga ega va shuning uchun ham algoritm hisobiga qo'shilmaydi.

E'tiboringizni yana bir narsaga qaratamiz. Keltirilgan misollarda quyidagi jumlar bor: «Ko'chadan o'tish» (ariqdan yoki dengizdan emas), «Eni 6 metr va bo'yi 10 metr bo'lgan joyni» (kilometr emas), «eni 12 santimetr va bo'yi 25 santimetrli g'ishtlar» (eni 5 santimetr va bo'yi 100 santimetrli g'ishtlar emas), « $x^2+x+1=0$ kvadrat tenglama» ($x^2-1=0$ kvadrat tenglama emas). Bu jumlar va qavs ichida yozilganlarni taqqoslasangiz, olinadigan natija shu jumalardagi «qiymat»larga chambarchas bog'liq ekanligini tushunasiz. Agar bu «qiymatlar» o'zgarsa, masalan, qavs ichidagilarga, olingadigan natija umuman boshqacha bo'lishini ko'rish qiyin emas. Qiymat so'zini qo'shtirnoq ichiga olganimizga sabab, siz doimo qiymat so'zini faqat sonlar bilan bog'lab o'rganib kelgansiz. Lekin, bilingki, biror masala uchun qiymat har xil turdagi obyektlar bo'lishi mumkin ekan. Demak, har bir algoritmning natijasi avvaldan berilgan, ya'ni boshlang'ich qiymatlarga bog'liq bo'lar ekan. Boshlang'ich qiymatlar turli natijalarga olib kelishiga yana bir hayotiy misolga o'zingiz javob bera olasiz: sizga va mehmonga kelgan do'stlaringizga pishiralayotgan palovga 20 gramm tuz solish o'rniga 200 gramm tuz solishsa natija bir xil bo'ladimi?

Har bir algoritm — bu amallarni belgilovchi qoida bo'lib, ularning zanjiri natijasida biz boshlang'ich qiymatlardan izlangan natijaga kelamiz. Bunday amallar zanjiri algoritmik jarayon, har bir amal — algoritmning qadami deb ataladi.

Yana boshlang'ich qiymat va natija bo'lishi mumkin bo'lgan narsalarning tahliliga qaytamiz. Ko'rdikki, har bir algoritm uchun boshlang'ich qiymatlarning turli hollarini tanlash mumkin. Masalan, g'ishtlar masalasi algoritmi uchun boshlang'ich qiymatlarni tavsiflashda «santimetr» so'zlarini «uzunlik o'lchovlari» kabi tushunish mumkin. Bu holda hosil bo'ladigan natijaning miqdori o'zgaradi, xolos. Ko'p algoritmlar boshlang'ich qiymatlarning turli hollari uchun o'z kuchini saqlab qoladi. Qo'shish algoritmini ixtiyoriy natural sonlar jufti uchun qo'llash mumkin. Avval aytib o'tilgan algoritmlarni aniqlangan bu xossasi (ularni boshlang'ich qiymatlar-

ning juda ko'p sondagi hollariga qo'llash mumkinligi) **ommaviylik** deb ataladi. Yuqorida keltirilgan « $ax^2+bx+c=0$ ($a\neq 0$) ko'rinishidagi kvadrat tenglamani yechish» algoritmi ixtiyoriy a , b , c haqiqiy sonlar uchun natija beradi, ya'ni algoritmning ommaviylik xossasi o'rinli ekan. Algoritmnlarni juda ko'p xususiy hollarda ishlashi shunday o'ta muhim va ahamiyatli xossa, shu sababli ancha vaqtgacha uni algoritmning ilmiy ta'rifiga kiritilishi shart deb hisoblandi. Bu ko'pgina qoidalarni fan sohasidan chiqarib tashlar (ularni «oz miqdordagi» ommaviyligi tufayli) va ham ilmiy tadqiqotni, ham ularning natijasini amaliyotda qo'llashni qiyinlashtirar (balki bu ilmiy bo'lmagan hol bo'lsachi?), bu esa jiddiy noqulaylikka sabab bo'ladi.

Boshlang'ich qiymatlarning faqat bitta – yagona holiga qo'llash mumkin bo'lgan algoritm ham katta ahamiyat kasb etadi. Ularga turli xil avtomatlardan (masalan, agar aniq bir tangaga moslangan gazeta sotadigan avtomat, yoki telefon-avtomat) foydalanish algoritmnlari tegishlidir, aniq bir joydan boshlanadigan va belgilangan joyga olib boradigan yo'nalish bo'yicha borish algoritmi va boshqalar.

«Ommaviylik» atamasining mavhumligi mashhur, ba'zan uyum paradoksi deb ataluvchi, Evbulid paradoksi orqali tasdiqlanadi. Paradoks mazmunini o'zimizga savol berib va javobini berib tezda aniqlab olishimiz mumkin. Bitta tosh – uyummi? Yo'q. Ikkita tosh-chi? Yana yo'q. Uchtasi-chi? Oxir-oqibat, biz yoki uyum mavjud emas degan xulosaga, yoki shunday sondagi toshlar to'plami borki, undan bitta oshsa uyum hosil bo'lishiga olib keladi, deb e'tirof etishga majbur bo'lamiz. U yoki bu fikr ham haqiqatga ziddir va bu uyum atamasining mavhumligining natijasi bo'lib hisoblanadi. Nima bo'lganda ham ommaviylik xossasidan oddiygina «bo'yin tovlash» mumkin emas. Uni ta'riflashni ozgina o'zgartirish hamda shuning asosida yuqorida aytib o'tilgan mavhumlikni yo'qotish zarur.

«Ommaviylik» atamasiga qanday mazmun kiritish kerak? Bu savolga javob shunday. Har bir algoritm uchun biror-bir obyektlar (narsalar, buyumlar va hokazo) sinfi mavjud va ularning barchasi boshlang'ich qiymat sifatida olinishi mumkin, deb hisoblash zarur. Manfiymas butun sonlarni qo'shish algoritmi uchun manfiymas butun sonlarning barcha juftligi; avtomatdan «Toshkent oqshomi» gazetasini sotib olish algoritmi uchun – yagona obyekt – tanga shunday sinf bo'la oladi. Algoritmning ommaviyligi – mos sinfn ing barcha obyektlarini qo'llash mumkinligidir, ya'ni, ularni qandaydir

miqdorining (chekli yoki cheksiz) qo'llash mumkin emasligi emas. Mumkin bo'lgan, ya'ni joiz obyektlarni miqdori chekli yoki cheksizligi, yoki miqdori nolga teng bo'lishi — bu shu sinfning xususiyatidir.

Agar algoritm yordamida joiz boshlang'ich qiymat asosida izlangan natijani olish mumkin bo'lsa u holda **algoritmni joiz boshlang'ich qiymatga qo'llash mumkin** deyiladi. Agar boshlang'ich qiymat joiz bo'lsa ham natija olish mumkin bo'lmasa, u holda unga **algoritm qo'llash mumkin emas** deyiladi.

Endi joiz boshlang'ich qiymatlar sinfi qanday ekanligini ko'rib chiqamiz. Boshlang'ich qiymatlar ba'zan narsa yoki buyumlar, sonlar ekanini ko'rdik. Bu fikr olingan natijalar uchun ham o'rinli. Bu narsalar orasidagi umumiylik nimada? Algoritm — bu qoidalar va demakki, ular qandaydir tillarda ifodilangan, degan fikrni e'tiborga olsak, bu umumiylik ko'rinadi. Bir necha marta bu qoidalarni aniq bajarilishi qanchalik muhim ekanligi haqida gapirib o'tdik.

Lekin bunday aniq bajarilishi boshlang'ich qiymatlar (ular bilan birga izlangan natijalar ham) biror-bir tilda, balki yanigisida, batamom tavsiflanishga imkon bersagina mumkin. Bu holda har bir boshlang'ich qiymatga, har bir oraliq natijaga va nihoyat, izlangan natijaga qandaydir gap mos keladi. Yana, mazkur gapning «Mazmun»i bir qiymatli bo'lishi zarur.

Matematikada ko'pincha maxsus usul qo'llanadi. Bu usul shundan iboratki, biror-bir obyekt boshqa tabiatli obyekt bilan almashtiriladi, bunda yangi obyektlarga birlamchilari bilan bir qiymatli mos bo'ladi. Ko'rilyotgan holda boshlang'ich qiymatlar tilining gaplari bilan boshlang'ich qiymatlarning o'zi orasida bir qiymatli moslik mavjud. Shu sababli, algoritmni matematik ta'riflashda boshlang'ich qiymatlar va izlangan natijalar tilning gaplari deb hisoblanishi mumkin.

Bunday almashtirish amaliyot nuqtayi nazaridan mumkinmi? Albatta, mumkin. Chunki, algoritmnining o'zida boshlang'ich qiymatlar emas, ularning **nomi**, jarayonni bajarish uchun esa amallar va hosil bo'ladigan natijalarning **nomini** bilish yetarli.

Keltirilgan usul algoritm ta'rifini tor ma'noda bo'lishiga olib keladi, deyish mumkin. Bunday fikr asoslidir. Lekin bu torayish muhim emas, chunki u algoritmlar beradigan imkoniyatlarni kamaytira olmaydi.

Bu kabi yondashish boshlang'ich qiymatlar va natijalar turlarini nisbatan kamaytiradi, ammo ular avvalgidek turli fizik

tabiatga ega bo'lishi mumkin, lekin biz uchun bu, ularni nazariy qaraganimizda, turli tillardagi gaplar kabidir. Narsalarning turlanishini biz tillarning turlanishiga keltirdik. To'g'ri, tillar ham kam emas. Ularni cheksiz to'plam (faqat mavjudlari emas, balki mavjud bo'lishi mumkin bo'lganlari ham, ya'ni mumkinlari ham) deb hisoblash mumkin. Lekin har bir algoritm faqat ikkita til bilan bog'langan: bittasida u ta'riflangan, ikkinchisining gaplari boshlang'ich qiymatlar hollarini uning uchun mumkin bo'lganlaridir. Birinchi tilni, odatda, **algoritmik til** deb, ikkinchisini — **operandlar tili** deb atashadi. **Operandlar** deb shunday obyektlarga aytiladiki, ular ustida algoritm talab qilgan amallar bajariladi. Operandlar tilining barcha gaplari joiz deb hisoblanadi, bu tilga tegishli bo'lmagan biror-bir belgilar birikmasi ta'rif bo'yicha joiz emas.

Algoritmni tasvirlash usullari

Algoritmnlarni tasvirlashning turli usullaru mavjud. Quyida algoritmnlarni tasvirlashning keng tarqalgan usullarini ko'rib chiqamiz.

1. Algoritmning so'zlar yordamida ifodalanishi

Avval keltirilgan bir qator misollar inson og'zaki nutqida qo'llaniladigan so'zlar orqali ifodalangan edi (masalan, ko'chadan o'tish algoritmi, g'ishtlar sonini hisoblash algoritmi). Algoritmning bunday tasvirlash usulida ijrochi uchun ko'rsatma jumlarlar orqali ko'rsatma shaklida beriladi. Qo'llanmada, asosan, shu usuldan foydalanamiz.

2. Algoritmning formulalar yordamida ifodalanishi

Bu usul matematika, fizika, kimyo va biologiya kabi fanlarda ko'plab qo'llaniladi. Yodingizda bo'lsa, so'zlar yordamida ifodalangan g'ishtlar sonini hisoblash algoritmini formula orqali ifodalagan edik. Formuladagi «+», «-», «x», «:» kabi arifmetik amallarning tartibiga rioya qilgan holda bajarilishi ham algoritmgga misol bo'ladi. Avval berilgan « $ax^2 + bx + c = 0$ ($a \neq 0$) ko'rinishidagi kvadrat tenglamani yechish» algoritmining quyidagi formula orqali ifodasi bilan tanishsiz:

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

Bilasiz, formuladagi amallar ma'lum bir tartib bilan bajarilishi shart.

3. Algoritmning jadval yordamida ifodalanishi

Algoritmning bu ko'rinishda berilishi ham sizga tanish. Masalan, matematikada qo'llanib kelinayotgan Bradis jadvali deb nomlangan to'rt honali matematik jadval, lotareya yutuqlar jadvali, Mendeleyev kimyoviy elementlar jadvali. Bunday jadvallardan foydalanish ma'lum bir algoritm qo'llashni talab etadi.

Biror funksiyaning grafigini chizish uchun ham funksiyaning argument qiymatlariga mos qiymatlar jadvalini hosil qilamiz. Bu ham algoritmning jadval ko'rinishiga misol bo'ladi.

4. Algoritmning grafik shaklda ifodalanishi

Algoritmning bu ko'rinishda ifodalanishi matematikada chizilgan grafik, kerakli uyni oson topish uchun dahalarda o'rnatilgan uylarning joylashish sxemasi, avtobuslarning yo'nalish sxemasi orqali sizga tanish.

Algoritmash asoslarini o'rganishning yana bir qulay grafik shakli — **blok-sxema** usulidir. Blok-sxemalar bir yoki bir nechta buyruq yoki ko'rsatmani aks ettiruvchi maxsus geometrik shakllar — bloklardan tashkil topadi. Bloklar yo'nalish chiziqlari orqali tutashiriladi.

	algoritmning boshlanishini va tugallanganligini bildiradi
	ma'lumotlarni kiritish yoki chiqarishni bildiradi
	oddiy harakatni, ya'ni qiymat berish yoki tegishli ko'rsatmalar berishni bildiradi
	shart tekshirilishini bildiradi
	yordamchi algoritmgga murojaatni bildiradi
	blok-sxemadagi harakat yo'nalishini bildiradi
	qiymat berish ko'rsatmasi



Cho'lpon

NMIU

**Cho'lpon nomidagi
nashriyot-matbaa ijodiy uyi**

ISBN 978-9943-05-439-4



9 789943 054394