

5- Mavzu: Umumiy algoritmlar nazariyasi

1. Algoritmlar nazariyasi fani maqsadi va vazifalari

Bizgacha etib kelgan intuitiv ma'nodagi algoritm eramizdan avvalgi III-asrda Evklid tomonidan taklif qilingan. Ushbu algoritm juda mashhur bo'lib, XX-asr boshlarigacha «algoritm» so'zining o'zi «Evklid algoritmi» ma'nosida ishlatilib kelindi. Boshqa matematika masalalarni bosqichli echishni tasvirlash uchun esa «usul» so'zidan foydalanilgan. Zamonaviy algoritmlar nazariyasi rivojida boshlang'ich nuqta deb, nemis matematigi Kurt Gyodelning ilmiy ishini ko'rsatib o'tish mumkin. Ushbu ishda ba'zi matematik muammolarni qaysidir sinfga taalluqli algoritmlar yordamida hal etib bo'lmasligi ko'rsatib berilgan.

1936 yilda Algoritmlar nazariyasi bo'yicha birinchi fundamental ilmiy ishlarni bir-biridan alohida tarzda Alan Tyuring, Aloiz Chyorch va Emil Postlar e'lon qildilar. Ular tomonidan taklif etilgan Tyuring mashinasi, Post mashinasi va Chyorchning λ -hisoblanuvchanlik usuli algoritm formalizmining ekvivalent shakllaridir. Ular tomonidan taklif etilgan tezislar algoritm intuitiv tushunchasi va formal tizimlarning ekvivalentligini ta'kidlab berdi. Algoritmik echimsiz muammolarning formulirovkasi va isboti ushbu ishlarning muhim natijasi bo'ldi. 1950- yillarda Algoritmlar nazariyasi rivojlanishiga rus matematiklari Kolmogorov va Markovlar o'z hissalarini qo'shdilar. 60-70-yillarga kelib Algoritmlar nazariyasi fanida quyidagi mustaqil yo'nalishlar ajralib chiqdi:

- Klassik algoritmlar nazariyasi (formal tillar terminlarida masalalarni ifodalash, echimli masala tushunchasi, 1965 yilda Edmonds tomonidan ta'riflangan PqNP muammosi, NP to'liq masalalar sinfining ochilishi va tekshirilishi);
- Algoritmlarning asimptotik analizi nazariyasi (asimptotik baholash usullari, algoritmlarning murakkabligi, algoritmlarni baholash kriteriyalari va h.k.z.). Ushbu yo'nalish rivojiga Knut, Axo, Xopkroft, Ulman, Karp kabi olimlar o'z hissalarini qo'shdilar;
- hisoblash algoritmlarining praktik analizi nazariyasi (algoritmlarning mehnattalabligi oshkor funksiyasini topish, funksiyalarning chegaraviy analizi, ratsional algoritmlarni tanlash metodikasi). Ushbu yo'nalish rivojlanishiga sabab bo'lgan.

2. Algoritmlar nazariyasining maqsadi va vazifalari

Algoritmlar nazariyasi turli yo'nalishlarining yutuqlarini umumlashtirib, uning maqsadi va vazifalarini ko'rsatib o'tish mumkin:

- Algoritm tushunchasini formallashtirish va formal algoritmik tizimlarni tekshirish;
- Bir qator masalalarning algoritmik echimsizligini formal isbotlash;
- Masalalar klassifikatsiyasi, murakkablik sinflarini aniqlash va tekshirish;
- Algoritmlar murakkabligining asimptotik analizi;

- Rekursiv algoritmlarni tekshirish va analiz qilish;
- Algoritmlar qiyosiy analizi uchun mehnattalablik oshkor funksiyasini topish;
- Algoritmlar sifatini qiyosiy baholash kriteriyalarini ishlab chiqish;

3. Algoritmlar nazariyasi fani natijalarining amaliy qo'llanilishi

Algoritmlar nazariyasidan olingan nazariy natijalardan amalda anchayin keng foydalanilmoqda. Bunda ikki aspektni alohida ko'rsatib o'tish mumkin:

Nazariy aspekt: qandaydir masalani tekshirish natijasida "Ushbu masala prinsipial jihatdan algoritmik echimlimi", degan savolga javob berish imkoniyati mavjud. Algoritmik echimsiz masalalar Tyuring mashinasi to'xtashi masalasiga olib kelinishi mumkin. Algoritmik echimli masalalar uchun esa, ushbu masalaning NP to'liq masalalar sinfiga mansubligi muhim ikkinchi nazariy savol bo'lib hisoblanadi.

Amaliy aspekt: Algoritmlar nazariyasi usullari quyidagi vazifalarni bajarishga imkon beradi:

- Berilgan masalani echish algoritmlari to'plamidan eng ratsional algoritmni tanlash;
- Murakkab masalalarni echish algoritmlarini vaqt jihatidan baholash;
- Kriptografik metodlar uchun muhim bo'lgan masala echimini ma'lum vaqt oralig'ida olib bo'lmasligini ishonchli baholash;
- Praktik analiz asosida axborotlarni qayta ishlash sohasidagi masalalarni echish effektiv algoritmlarini ishlab chiqish va rivojlantirish;

4. Algoritm tushunchasini formallashtirish

Inson o'zining barcha faoliyat sohalarida, jumladan axborotlarni qayta ishlashda ham masalalarni echishning turli usul va vositalari bilan to'qnashadi. Ular pirovard natijaga erishish uchun bajariladigan harakatlar tartibini aniqlaydi. Buni intuitiv ma'nodagi algoritm tushunchasi deb qarashimiz mumkin. Ushbu tushunchaga qo'yiladigan ba'zi talablar esa algoritmni noformal aniqlash imkonini beradi:

Algoritm - qaysidir tilda berilgan masalani echish uchun bajariladigan boshlang'ich berilganlar ustida bajariladigan amallarning chekli ketma-ketligi.

Algoritm tushunchasiga bitta konkret ta'rif berishning imkoni bo'lmadi.

Algoritmlar nazariyasida algoritmning turli formal ta'riflari kiritilgan bo'lib, ularning ekvivalentligi isbotlangan.

Algoritm – bu boshlang'ich berilganlardan izlangan natijaga olib keluvchi hisoblash jarayonini aniqlovchi aniq ko'rsatmalardir.

Algoritm tushunchasining turli ta'riflari bir qator talablarga javob berishi kerak:

- algoritm chekli sondagi elementar bajariluvchi ko'rsatmalardan iborat bo'lishi kerak;
- algoritm chekli sondagi qadamlardan iborat bo'lishi kerak;
- algoritm barcha boshlang'ich berilganlar uchun umumiy bo'lishi kerak;

- algoritm to'g'ri echimga olib kelishi kerak.

5.Intuitiv algoritm tushunchasi

Hisoblash mashinasining ishi algoritmlarni bajarishdan iborat bo'ladi. Shuning uchun hisoblash mashinalarining umumiy imkoniyatlari qaysi muammo-masalalarni algoritm sifatida tasvirlash mumkinu, qaysilarini mumkin emasligiga bog'liq bo'ladi. Matematikaning eng asosiy tushunchalaridan biri bo'lgan algoritm tushunchasi hisoblash masalalari paydo bo'lganidan ancha oldin vujudga kela boshlagan edi. Asrlar davomida kishilar intuitiv algoritm tushunchalaridan foydalanib kelganlar. Bu tushunchani shunday ta'riflash mumkin:

Algoritm – bu qoidalarning qat'iy va chekli sistemasi bo'lib, ba'zi ob'ektlar ustida bajariladigan amallarni aniqlaydi va chekli qadamdan keyin qo'yilgan maqsadga olib kelishni ta'minlaydi.

6.Algoritm ob'ekti va alfaviti

Intuitiv algoritm tushunchasi b'ekti sifatida ixtiyoriy narsa olinishi mumkin edi. Tabiiyki, ishni ob'ekt tushunchasini formallashtirishdan boshlash kerak edi. Hisoblash algoritmlarida ish ob'ektlari sifatida sonlar olinadi. Shaxmat o'yini algoritmidagi ob'ektlar bu – shaxmat figuralari va pozitsiyalaridir. Ishlab chiqarish jarayonlarini algoritmlashtirishda ob'ektlar sifatida priborlar ko'rsatishlari va boshqaruv tugmalari olinadi. Bunda klavishlarning shunday bosilish tartibi aniqlanishi kerakki, ishlab chiqarish jarayoni eng yaxshi bo'lsin. Bu algoritmlar turli-tumanligiga bir necha misol holos. Ammo barcha hollarda real olam ob'ektlari bilan emas, ularning tasviri bilan ish ko'riladi

Belgilarni harflar deb, ularning to'plami esa alfavit deb ataladi. Umumiy holda harflar sifatida ixtiyoriy belgilar olinishi mumkin. Bunda ular o'zaro turli xil bo'lishi va ularning chekliligi talab eliladi. Demak, harflar bu – ixtiyoriy belgilar; alfavit esa – o'zaro turlich bo'lgan harflarning chekli to'plamidir.

Qandaydir alfavitdagi ixtiyoriy harflarning chekli ketma-ketligi ushbu alfavitdagi so'z deb ataladi. Masalan, qo'shishi algoritmidagi + belgisi bilan ajratilgan qo'shiluvchilarning tasvirlarini yig'indini ifodalovchi tasvirga aylantiradi. Bunda so'zlardagi harflarning tartibi juda muhim bo'lib alfavitdagi tartibi esa muhim emasdir. $\{A, B\}$ va $\{B, A\}$ alfavitlar bir xildir, ammo AB va BA so'zlar har xildir.

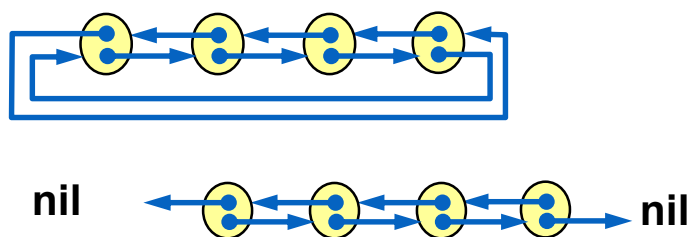
So'zlardagi harflar soni so'zning uzunligi deyiladi. Shunday qilib, real olam ob'ektlarini turli alfavitlardagi so'zlar bilan ifoda etish mumkin. Bu esa algoritmlarning ish ob'ektlari sifatida faqat so'zlar qatnashishi mumkin deb aytish imkonini beradi. Bundan shunday xulosaga kelish mumkin:

Agar har bir alfavitdan 2 ta belgili alfavitga o'ish va kodlangan belgilarni so'zlarga aylantirish imkoni bo'lsa, ixtiyoriy algoritmni 0 va 1 lar ustidagi amallar ketma-ketligiga keltirish mumkin bo'ladi.

Algoritm tushunchasi juda qadim zamonlardan shakllanib kelgan. Shunga qaramay, XX asrning yarmiga qadar matematiklar bu ob'ekt haqida intuitiv qarashlarga qanoatlanib kelganlar. Algoritm termini matematiklar tomonidan faqat konkret masalalarni echish bilan bog'liq holda olinar edi. XX asr boshida matematika asoslarida vujudga kelgan qarama-qarshiliklar va muammolar ularni etishga qaratilgan turli kontsepsiyalar va oqimlarning vujudga kelishiga olib keldi. 20- yillarga kelib, effektiv hisoblash masalalari ko'ndalang bo'ldi. Algoritm tushunchasining o'zi matematik tadqiqotlar ob'ekti bo'lib qolganligi uchun aniq va qat'iy ta'rifga muxtoj edi. Bundan tashqari EHM lar asrini yaqinlashtiruvchi fizika va texnikaning rivojlanishi ham shuni taqozo etar edi. XX asr boshlarida matematiklar ba'zi ommaviy masalalar algoritmik echimga ega emas degan xulosaga kela boshladilar. Biror bir ob'ektning mavjud emasligini qat'iy isbotlash uchun esa, ushbu ob'ektning aniq ta'rifiga ega bo'lish kerak edi. Uzluksizlik, egri chiziq, sirt, uzunlik, yuza, xajm va boshqa shu kabi tushunchalarni konkretlashtirish zarurati tug'ilganda xuddi ana shunday holat vujudga kelgan edi.

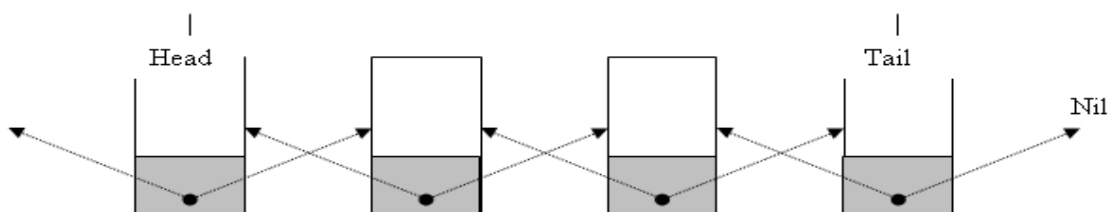
7. Berilganlarning dinamik tuzilmasi tushunchasi

Algoritm lash va dasturlashda operativ xotirada ma'lumotlarning dinamik(o'zaruvchan)to'plamlarini saqlashga ehtiyoj tug'iladi. Ma'lumotlarning dinamik to'plami deganda algoritm bajarilishi jarayonida o'zgaruvchi to'plam tushuniladi. Ko'p hollarda dinamik to'plamlarni saqlashda massivlardan foydalanish noqulaydir. Chunki massivlarning belgilangan o'zgarmas xajmi bilan dinamik to'plamning o'zgaruvchan xajmi bir-biriga to'g'ri kelmaydi. Bunda massiv xajmini dinamik ob'ektning maksimal xajmiga teng qilib olishga to'g'ri keladi. Ammo bunday usul operativ noeffektiv foydalanish muammosini tug'diradi. Shu sababli dinamik ma'lumotlarni saqlash va qayta ishlash uchun maxsus ma'lumot tuzilmalaridan foydalaniladi. Ma'lumotlarning dinamik tuzilmalari chiziqli va tarmoqli turlarga bo'linadi.



8. Ma'lumotlarning chiziqli tuzilmalari

Ro'yxat – bu berilganlarning ketma-ket tashkillashtirilgan strukturasi. Chiziqli ro'yxatlarning massivlardan farqi shundaki, ular dastur bajarilishi jarayonida o'z xajmini o'zgartirish imkoniyatiga ega. Binobarin ro'yxatlarning xajmi oldindan aniqlanmaydi. Chiziqli ro'yxatni zanjir qismlari ko'rinishida tasvirlash mumkin:



Ma'lumotlarning chiziqli tuzilmalari **ro'yxatlar, ctklar, navbatlar va siklik ro'yxatlar** deb ataluvchi turlarga bo'linadi. Ushbu ma'lumot tuzilmalari elementlari zanjir hosil qilib, har bir element ko'rsatkich yordamida o'zidan oldingi yoki o'zidan keyingi elementlarga ko'rsatadi. Ro'yxat boshidagi element **head** (bosh), oxiridagi element **tail** (dum) deb ataladi

9.Ro'yxatlar

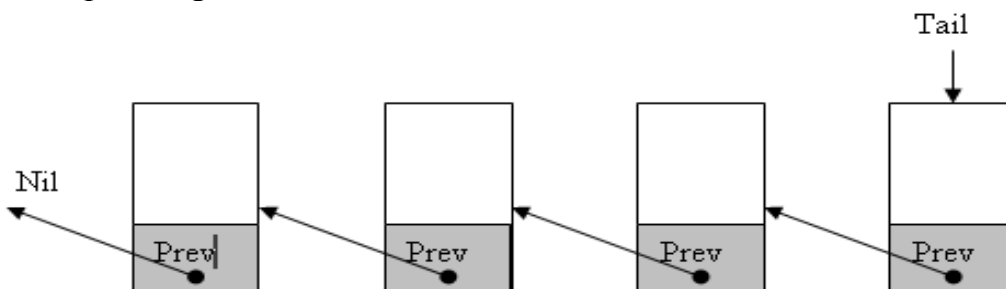
Bunda to'g'ri to'rtburchaklar bilan ro'yxat elementlari ifodalangan. Elementlar ikki qismdan iborat: yozuv sohasi(ma'lumotlar) va adres sohasi. Adres sohasi keying yoki oldingi elementga murojaat ko'rsatkichini saqlaydi. Ro'yxatning barcha elementlariga murojaat qilish uchun ro'yxat boshi yoki oxirining adresi kifoya qiladi. Ro'yxatlar bir yo'nalishli va ikki yo'nalishli bo'lishi mumkin.

10.Steklar

“Oxirgi kelgan birinchi ketadi” (LIFO –Last in First Out), tamoyili asosida tashkil etilgan ro'yxatlar steklar deb ataladi. Stekda ixtiyoriy lementlarni o'chirish man qilinadi. Bu erda faqat oxirgi elementni o'chirishga ruxsat etiladi. Steklar quyidagi tarzda tashkil etiladi:

Stekning har bir elementi (birinchi elementdan tashqari) o'zidan oldingi elementga murojaatga ega bo'ladi;

- Stekka murojaat uchun oxirgi element adresi qo'llaniladi (Tail);
- Yangi element stek oxiriga joylashtiriladi;
- Faqat oxirgi element o'chiriladi.



11.Navbatlar

“Oldin kelgan – oldin ketadi”. FIFO (First in First Out), tamoyili asosida tashkil etilgan ro'yxatlar navbatlar deb ataladi. Navbatda ixtiyoriy elementlarni o'chirish man etiladi. Faqat birinchi elementni o'chirishga ruxsat etiladi.

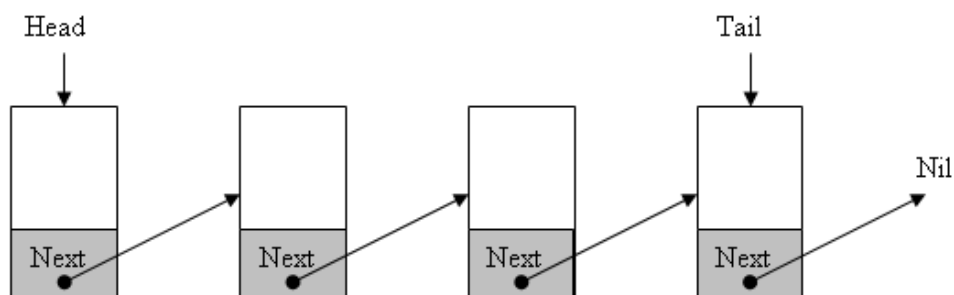
Navbatning har bir elementi (oxirgi elementdan tashqari) o'zidan keyingi elementga murojaatga ega bo'ladi;

- Stekka murojaat uchun bosh (Head); va oxirgi element adresi qo'llaniladi (Tail);
- Yangi element navbat oxiriga joylashtiriladi;
- Faqat birinchi element o'chiriladi

12. Ikki yo'nalishli siklik ro'yxatlar

Ro'yxat oxirgi elementi birinchi elementga murojaat qilsa, bunday ro'yxatlar **siklik** deb ataladi. Siklik ro'yxatlar quidagi tarzda tashkil qilinadi:

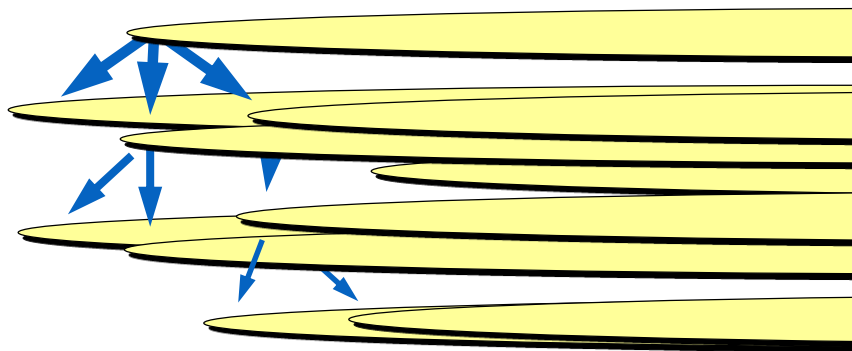
1. Oxirgi element birinchi elementga murojaat etadi;
2. Po'yxat elemenlariga murojaat etish uchun oxirgi element (Tail) va birinchi element (Head) (shart emas) adresidan;
3. Yangi element ro'yxat oxiriga joylashtiriladi;
4. Elementni o'chirish kalit bo'yicha amalga oshiriladi (avval kalit bo'yicha izlash amalga oshiriladi).
5. Siklik ro'yxatlarga izlash va o'chirish algoritmlarini soddalashtirish imkonini beruvchi "soxta" element kiritish qabul qilingan



13. Ma'lumotlarning nochiziqli tuzilmalari

Dinamik tuzilmalarning nochiziqli turiga **daraxtlar** va **graflar** kiradi. Bunda ma'lumotlar bog'lanish tarmoqlanuvchi tuzilishga ega bo'ladi.

Daraxtlar. Daraxt bu – ma'lumot tugunlari va ularni bir-biriga bog'lovchi yo'nalishli murojaatlar majmuasidir. Daraxtdagi ma'lumot tugunlari avlod va ajdod turlariga mansub bo'lishi mumkin. Bog'lanishlar chiqadigan tugun **ajdod**, bog'lanish kiradigan tugun **avlod** tugun deb ataladi Daraxtning barcha tugunlariga ajdod bo'lib, o'z ajdodiga ega bo'lmagan, faqat avlod tugunlarga ega bo'lgan tugun **daraxt ildizi** deb ataladi.



14. Boshlang'ich berilganlar va hisoblash jarayoni

Axborot tushunchasi asosiy ilmiy tushunchalarga kiradi, shuning uchun uni aniq intuitiv deb hisoblaymiz va bu tushunchani aniqlashtirib o'tirmaymiz. Biz kursimizda EHMda axborotga ishlov berishni ko'rib chiqamiz. Shuningdek, biz axborotning belgilar to'plami bilan ifodalanuvchi diskret ko'rinishi bilan chegaralanamiz, masalan, turli tildagi matnlar, sonlar. Agar axborotning boshlang'ich ko'rinishi boshqa ko'rinishda (masalan, rasmdagi tasvir) berilgan bo'lsa, unga EHM da ishlov berish uchun diskret ko'rinishga almashtiramiz (bizning misolda ranglarni raqamlash, rasmni kichik kvadratlarga bo'lish va har bir kvadrat qanaqa rangda ekanligini yozib qo'yish mumkin). Axborotga ishlov berish quyidagi sxema bo'yicha amalga oshiriladi:

BOSHLANG'ICH AXBOROT → IZLANAYOTGAN AXBOROT

ya'ni, boshlang'ich axborotdan izlanayotgan axborot topiladi. Axborotga ishlov berishni uni bir formadan boshqasiga tarjima qilish sifatida qarash mumkin.

15. Algoritmning axborotlarni qayta ishlash jarayoni

Algoritm va hisoblash jarayoni orasidagi farqni ko'rish qiyin emas. Masalan, algoritmda bir marta uchragan amal bir necha marta bajarilgan bo'lishi mumkin va hisoblash jarayonida bir necha marta ifodalanishi mumkin. Bu amallarga misol sifatida 3- yoki 5- yoki 6-amallarni keltirishimiz mumkin. Bizning misolda qoida bo'yicha bir turdagi amallar turli berilganlar ustida bajariladi. Biroq, bu berilganlar bir xil nomda tasvirlanishi mumkin.

Algoritm bo'yicha har doim ham hisoblash jarayonini oldindan aytib bera olmaymiz. Masalan, hisoblash jarayonida amallar soni qanchalik ko'p bo'lishini oldindan aytish qiyin. Algoritmni bajarish №1 amaldan boshlanadi. Algoritmda har doim keyingi bajariladigan amal aniqlangan bo'ladi. Yashirin holda u keyingi nomer bilan belgilangan amal hisoblanadi. Agar algoritmdagi amallar tartibi nomerga mos tushmasa, u o'tish amali yordamida ko'rsatiladi. To'xtatish amalidan so'ng algoritm bajarilishi to'xtatiladi. Shunday qilib, navbatdagi amal bir qiymatli aniqlangan. Determinallashgan deb nomlanuvchi bu algoritmning xossasiga ko'ra, boshlang'ich

berilganlar uchun hisoblash jarayoni har doim aniqlangan bo'ladi. Shunday qilib, bir xil boshlang'ich berilganlar uchun hisoblash jarayoni ham bir xil bo'ladi.

Amallar tushunchasini ko'rib chiqamiz. Bizning misolda amalning 2 ko'rinishi mavjud:

- berilganlar ustida amallar (taqqoslang, ayiring, qo'ying);
- hisoblashlar qatorini boshqaruvchi amallar (agar u holda; o'ting, to'xtating).

16. Algoritmning murakkabligi

- Bir xil turdagi masalalar sinfini echish uchun bir nechta turli algoritmlar mavjud. Ular asosida vujudga kelgan hisoblash jarayonlari amallar to'plami va miqdori bilan farq qiladi. Hisoblash jarayonidagi amallar miqdori algoritmnining muhim tomonlaridan biri hisoblanadi, chunki u algoritmni bajarish uchun kerak bo'lgan bajaruvchining vaqti va resurslarini aniqlaydi.
- Algoritmning murakkabligi deb, hisoblash jarayonida boshlang'ich berilganlar uchun berilganlar to'plami asosida vujudga kelgan algoritmdagi amallar miqdoriga aytiladi.
- Bir turdagi masalalar sinfini echish uchun turli murakkablikdagi turli algoritmlar mavjud.

17. Algoritmning asosiy xossalari

- ommaviylik;
- potentsial amalga oshuvchanlik (tugallanganlik, murakkablik);
- determinallashtirilganligi;
- barcha bajaruvchilarning tushunishi.

Ommaviylik xossasi ma'lum masalalar sinfi nusxalarining echimi uchun algoritmni qo'llash mumkinligini o'zida namoyon etadi. Masalan, $2 \times 2 = 4$ algoritmi emas, chunki ommaviylik xossasi yo'q, ya'ni yagona masala echilyapti, boshlang'ich berilganlar qat'iy berilgan va o'zgartirilishi mumkin emas.

Potentsial amalga oshuvchanlik xossasi, ixtiriy boshlang'ich berilganlarda algoritmi hisoblash jarayonini vujudga keltirishini ifodalaydi. Shunday qilib, ixtiyoriy mumkin bo'lgan kiruvchi berilganlar to'plamida algoritmi murakkabligi cheksiz.

Determinallashtirilganlik xossasi algoritmi natijasi ixtiyoriy bo'lishi mumkin emas, u doim kiruvchi berilganlar bilan aniqlanishini ta'kidlaydi. Shuningdek, determinallashtirilganlik xossasi vujudga kelgan algoritmi hisoblash jarayonida qat'iy berilgan masalaga o'tish jarayonida ixtiyoriy tasodif tushirib qoldiriladi. Shunday

qilib, algoritm va boshlang'ich berilganlar faqat ular axborotga ishlov berish jarayonini aniqlaydi.

Bajaruvchilarning bir qiymatli tushunish xossasi algoritm ifodasida noaniqliklar bo'lmasligini ko'rsatadi. U bajaruvchiga tushunarli bo'lishi shart va faqatgina bajaruvchi amalga oshira oladigan amallarni o'zida mujassamlashtirishi kerak. Nisbatan murakkab masalalarni echishda algoritmdan muayyan kompyuter tilidagi dasturga o'tish juda qiyin. Bunday bevosita o'tishda algoritmning alohida qismlari orasidagi bog'lanish yo'qoladi, algoritm tarkibining asosiy va muhim bo'lmagan qismlarini farqlash qiyin bo'lib qoladi. Bunday sharoitda keyinchalik aniqlash va to'g'rilash ancha vaqt talab qiladigan xatolarga osongina yo'l qo'yish mumkin. Odatda algoritm bir necha marta ishlab chiqiladi, ba'zan xatolarni to'g'rilash, algoritm tarkibini aniqlashtirish va tekshirish uchun bir necha marta orqaga qaytishga to'g'ri keladi. Algoritm ishlab chiqishning birinchi bosqichida algoritmni yozishning eng qulay usuli - algoritmni tuzim ko'rinishda ifodalashdir.

Dasturlash tilining kompyuter tiliga yaqinligi darajasini tariflash uchun til darajasi tushunchasi qo'llaniladi. Kompyuter tili 0 daraja deb qabul qilingan bo'lib, sanoq boshi hisoblanadi. Odamning tabiiy tili "eng yuqori darajadagi til" deb qaraladi. Kompyuterga bog'liq bo'lmagan tillar ham ikkita turga bo'linadi: birinchisi protseduraga mo'ljallangan tillar, ikkinchisiga - muammoga mo'ljallangan tillar. Protseuraga mo'ljallangan tillar turli masalalarni echish algoritmlarini (protseuralarni) tavsiflashga mo'ljallangan; shuning uchun ular ko'pincha oddiy qilib "algoritmik tillar" deb ataladi. Ushbu tillar echilayotgan masalalar xususiyatlarini to'la hisobga oladi va kompyuterning turiga deyarli bog'liq emas. Bu xildagi tillar tarkibi kompyuter tiliga qaraganda tabiiy tilga, masalan, ingliz tiliga yaqinroq.

Foydalanilgan adabiyotlar

1. Informatika va informatsion texnologiyalar, M.Aripov va boshqalar. Oliy o'quv yurti talabalari uchun darslik. Toshkent-2019 y.
2. Axborot texnologiyalari, M.Aripov va boshqalar. Oliy o'quv yurti talabalari uchun o'quv qo'llanma. Toshkent-2019 y.
3. Delphi tilida dasturlash asoslari, Sh.Nazirov. Toshkent-2018 y.