



ALGORITMLASH TAMOYILLARI VA ULARNING ZAMONAVIY DASTURLASHDAGI AHAMIYATI

Nabijonov Jasurbek Jaxongir o'g'li

Farg'ona viloyati O'zbekiston tumani

2-son texnikumi

Maxsus fan o'qituvchisi

Annotatsiya. Ushbu maqolada algoritmlash tamoyillari va ularning zamonaviy dasturlash jarayonidagi o'rni keng yoritiladi. Algoritm tushunchasi dastur yaratishning asosiy manbai sifatida ko'rib chiqilib, uni shakllantirishda qo'llaniladigan asosiy tamoyillar — ketma-ketlik, tanlash, takrorlanish, modullilik hamda samaradorlik tahlil qilinadi. Shuningdek, algoritm tuzishning muammolarni hal etishdagi ahamiyati, masalani formal ifodalash jarayonida algoritmik fikrlashning roli batafsil ochib beriladi. Maqolada algoritmlarning turli dasturlash tillaridagi realizatsiyasi, ularning texnologik rivojlanish bilan uzviy bog'liqligi ham misollar yordamida ko'rsatib berilgan. Zamonaviy dasturlashda optimallashtirilgan algoritmlardan foydalanish dastur tezligi, resurslardan samarali foydalanish va tizim barqarorligini ta'minlashdagi eng muhim omillardan biri sifatida baholangan. Tadqiqotda algoritmlash tamoyillarining sun'iy intellekt, ma'lumotlar tahlili va avtomatlashtirish jarayonlaridagi qo'llanilishiga ham e'tibor qaratilgan.

Kalit so'zlar: algoritm, algoritmlash tamoyillari, dasturlash, ketma-ketlik, tanlash, takrorlanish, samaradorlik, modullilik, optimallashtirish.

Zamonaviy axborot tizimlari, mobil ilovalar, veb-platformalar, sun'iy intellekt tizimlari va boshqa ko'plab texnologik yechimlar asosida aniq tuzilgan va puxta o'ylangan algoritmlar yotadi. Algoritmlar — bu kompyuter bajarishi uchun mo'ljallangan aniq, izchil va mantiqiy buyruqlar ketma-ketligidir. Dasturlashning qanchalik rivojlanganidan qat'i nazar, har bir dasturchi eng avvalo algoritmlash tamoyillarini chuqur bilishi zarur, chunki samarali algoritmlar yaratilmasdan turib,



yuqori sifatli dasturiy mahsulot yaratish imkonsiz. Mazkur maqolada algoritmlash tamoyillarining nazariy asoslari, ularning dasturlash jarayonida tutgan o'рни, zamonaviy texnologiyalar bilan bog'liqligi hamda optimallashtirish masalalari keng yoritiladi.

Algoritm — bu ma'lum bir masalani hal qilish uchun zarur bo'lgan ketma-ket bajariladigan harakatlar majmui. Har qanday algoritm quyidagi xususiyatlarga ega bo'lishi kerak: aniqlik, izchillik, tugallanish, umumiylik va samaradorlik. Aniqlik algoritmda har bir buyruqning tushunarli va aniq belgilanganligini bildiradi. Izchillik buyruqlarning mantiqiy ketma-ketligini ta'minlaydi. Tugallanish algoritmning ma'lum bir vaqtda yakuniy natijaga olib kelishini anglatadi. Samaradorlik esa algoritmning vaqt va xotira resurslaridan oqilona foydalanishini bildiradi. Ushbu xususiyatlar har qanday algoritmni to'g'ri tashkil etishda muhim omil bo'lib xizmat qiladi.

Algoritmlash tamoyillari

Dasturlashning asosiy tamoyillari uchta fundamental prinsipga tayanadi:

Ketma-ketlik tamoyili

Tanlash (shart) tamoyili

Takrorlanish tamoyili[1]

Bu tamoyillar deyarli barcha dasturlash tillari va algoritmik modellarning negizini tashkil etadi.

Ketma-ketlik tamoyili barcha buyruqlar qat'iy tartibda bajarilishini bildiradi. Bu eng sodda, lekin eng muhim tamoyil bo'lib, dastur oqimining asosiy yo'nalishini tashkil etadi. Dastur bajarilishida har bir buyruq avvalgisidan keyin bajariladi. Keng ko'lamli dasturlarda ham jarayonning katta qismi aynan ketma-ketlikka asoslanadi.

Tanlash tamoyili shartlarning bajarilishiga qarab turli yo'nalishlarni tanlash imkonini beradi. Masalan, “agar shart bajarilsa — birinchi blok, bajarilmasa — ikkinchi blok bajariladi” ko'rinishidagi operatorlar tanlashning amaliy ko'rinishidir. Bu tamoyil dasturga moslashuvchanlik va mantiqiy murakkablik qo'shadi.

Takrorlanish yoki sikl tamoyili ma'lum buyruqlarni bir necha bor bajarishni ta'minlaydi. Sikllar algoritmni soddalashtiradi va resurslardan unumli foydalanish



imkonini beradi. Takrorlanishning asosiy turlari: “for”, “while”, “do-while” sikllari bo‘lib, ular orqali takrorlanuvchi jarayonlar to‘liq avtomatlashtiriladi.[2]

Algoritmashning yana bir muhim tamoyili — modullilik. Modullilik dastur kodini alohida bo‘laklarga ajratib, har bir bo‘lakni mustaqil tarzda bajariladigan funksiyaga aylantiradi. Bu dastur tuzilishini soddalashtiradi, kodni qayta ishlashni osonlashtiradi va xatoliklarni kamaytiradi. Modulli yondashuv keng ko‘lamli dasturlarni samarali tashkil qilishda asosiy vositadir.

Zamonaviy dasturlashda faqat to‘g‘ri ishlaydigan algoritm yaratish kifoya qilmaydi. Eng muhim jihatlardan biri — algoritm samaradorligi. Bu algoritmning vaqt sarfi (Time Complexity) va xotira sarfi (Space Complexity) bilan baholanadi. Masalan, $O(n)$, $O(n^2)$, $O(\log n)$ kabi murakkablik ko‘rsatkichlari algoritmning samaradorlik darajasini ifodalaydi. Samarasiz algoritmlar katta hajmdagi ma’lumotlar bilan ishlaganda tizimni sekinlashtirishi yoki ortiqcha resurs sarflashga olib kelishi mumkin.[3]

Bugungi kunda dasturlash sohasining barcha yo‘nalishlari algoritmlarga tayanadi:

Sun’iy intellekt algoritmlari neyron tarmoqlarni o‘qitishda qo‘llanadi.

Ma’lumotlar tahlili va Data Science algoritmlar orqali katta ma’lumotlarni tartiblash, tasniflash va prognozlashni amalga oshiradi.

Kiberxavfsizlik sohasida shifrlash va himoya algoritmlari qo‘llaniladi.

IoT tizimlari algoritmik boshqaruv orqali real vaqt rejimida ishlaydi.

Robototexnika algoritmlarsiz harakat, nishonga olish yoki muvozanatni ta’minlay olmaydi.[4]

Bu jarayonlarning barchasi algoritmash tamoyillarining qanchalik muhimligini yana bir bor tasdiqlaydi.

Optimallashtirish — algoritmni yanada samarali qilish jarayonidir. U quyidagi yo‘llar bilan amalga oshiriladi:

kerakmas takrorlarni kamaytirish;

mantiqiy shartlarni qisqartirish;

samaraliroq ma’lumotlar strukturasi tanlash;



murakkablik darajasini pasaytirish.[5]

Optimallashtirilgan algoritmlar tezroq ishlaydi, kamroq resurs talab qiladi va tizimni barqaror qiladi.

Xulosa qilib aytganda, algoritmlash tamoyillari zamonaviy dasturlashning eng asosiy poydevori hisoblanadi. Kod qanchalik mukammal ko‘rinsa ham, uning asosi bo‘lgan algoritmlar ishonchli, samarali va puxta o‘ylangan bo‘lmasa, natija kutilgan darajada bo‘lmaydi. Bugungi raqobatbardosh texnologiyalar davrida har bir dasturchi algoritmik fikrlashni rivojlantirishi, tamoyillarni chuqur o‘zlashtirishi va samaradorlikka alohida e‘tibor qaratishi zarur. Maqolada ko‘rsatib o‘tilgan tamoyillar nafaqat dasturlash jarayonini soddalashtiradi, balki murakkab muammolarni hal etishda kuchli nazariy va amaliy asos yaratadi.

FOYDALANILGAN ADABIYOTLAR

1. Knuth, D. “The Art of Computer Programming”. Addison-Wesley, 2017. — 112–130-betlar.
2. Cormen, T. H., Leiserson, C. E., Rivest, R., Stein, C. “Introduction to Algorithms”. MIT Press, 2019. — 45–78-betlar.
3. Aho, A. V., Hopcroft, J. E., Ullman, J. D. “Data Structures and Algorithms”. Addison-Wesley, 2013. — 20–54-betlar.
4. Sedgewick, R., Wayne, K. "Algorithms". Pearson, 2021. — 90–140-betlar.
5. Lafore, R. “Object-Oriented Programming in C++”. SAMS Publishing, 2022. — 15–33-betlar.