

## ALGORITMLAR ISHLAB CHIQUISH METODLARI VA ULARNING TAHLILI

Hasanova Dilafroʻz Boʻronovna  
Buxoro davlat pedagogika instituti, "Aniq fanlar" kafedrasida o'qituvchisi

*Annotatsiya.* Hozirda axborot-texnologiyalar dasturlarni loyihalashda algoritmlarni sodda, tushunarli, xatosiz, aniq qadamlarda tezkor natija beradigan algoritm tuzish muhim omil hisoblanadi. Ushbu maqolada berilgan masalani algoritmini tuzish uchun loyihalash texnikasi asosida qadamlar ketma-ketligida koʻrib chiqish va turli metodlar bilan ishlab chiqish, tahlil qilish va algoritmlarning kompyuter qurilmalari samaradorligiga ta'siri ochib berilgan.

*Kalit soʻzlar:* algoritm, chiziqli, tartiblash, izlash, samaradorlik, Big O, blok-sxema.

## МЕТОДЫ РАЗРАБОТКИ АЛГОРИТМОВ И ИХ АНАЛИЗ

Хасанова Дилафруз Бурановна  
Бухарский государственный педагогический институт, преподаватель кафедры точных наук

*Аннотация:* В настоящее время при разработке программ информационных технологий важным фактором является создание алгоритмов, простых, понятных, безошибочных и дающих быстрые результаты на конкретных шагах. В данной статье представлен пошаговый обзор на основе методики проектирования создания алгоритма для заданной задачи и разработки различными методами, анализ и влияние алгоритмов на эффективность компьютерных устройств.

*Ключевые слова:* алгоритм, линейный, поиск, поиск, эффективность, Big O, блок-схема.

## ALGORITHM DEVELOPMENT METHODS AND THEIR ANALYSIS

Hasanova Dilafuz Buronovna  
Bukhara State Pedagogical Institute, Teacher of the «Exact Sciences» department

*Abstrakt.* Currently, in the design of information technology programs, it is an important factor to create algorithms that are simple, understandable, error-free, and give quick results in specific steps. In this article, a step-by-step review based on the design technique for creating an algorithm for the given problem and development with different methods, analysis, and the impact of algorithms on the efficiency of computer devices are revealed.

*Key words:* algorithm, linear, searting, search, efficiency, Big O, flowchart.

*Kirish.* Bugunda hammaga ayonki, axborot texnologiyalarning zamonaviyligi, samaradorligi va qulayligi har bir sohadan tortib ijtimoiy hayotgacha boʻlgan asosiy ustivorlikni belgilaydi. Garchi, buni anglasakda lekin bu qurilmalarning va dasturlarning asosida yotgan algoritmlar, ularning ishlash tartibida nimalar hisobga olinganligi toʻgʻrisida har doim ham bilavermaymiz. Buyuk Britaniyalik olim Devid Xarel oʻzining "Algoritmika:hisoblash ruhi" kitobida "Algoritmika bu kompyuter fanining bir boʻlimi emas. Bu informatika fanining oʻzagi..." deb taʼriflaydi. Algoritmishni oʻrganish nafaqat kompyuter fanlarini bilishga turtki beradi, shuningdek muammolarni hal qilish strategiyasini ishlab chiqishni ham oʻrgatadi. Berilgan masalaning dasturini tuzganda bu kompyuter parametrlariga qay darajada taʼsir qilishi va bu jarayonlarni qay tartibda foydali jihatlarini oshirish ishlab chiqilayotgan algoritmgaga bogʻliq. Villanova universiteti professori Anany Levitin "Algoritmlarning tahlili va turlari" ilmiy nashrida algoritmlar ikkita muqobil variantga koʻra tavsiylaydi; birinchisi asosan berilgan masalaning turiga qarab tavsiflaydi va ularni lineary(chiziqli), searting(tartiblash), searching(izlash) turlariga boʻlib oʻrganish boʻlsa, ikkinchi alternativa esa bu algoritmlarni loyihalash texnikasi asosida tashkil qiladi.

Bizning darslik va adabiyotlar birinchi variantga koʻra algoritmlarni oʻrganish va oʻrgatish keng tarqalgan boʻlib, buning ham oʻz navbatida ijobiy tomoni turli algoritmlarni samaradorligini taqqoslash darhol namoyon boʻladi va kamchilik sifatida loyihalash texnikasi sifatida muammo turlariga urgʻu berilishini aytib oʻtishimiz mumkin. Algoritmlarni loyihalash bugungi kompyuter fanlarini tezkorlik bilan rivojlanayotgan bir paytda eng yaxshi loyihalash usuli boʻlib, tuzuvchini aniq bir qolipga solib qoʻymasdan uning tanqidiy fikrlashini va yaratuvchanligini shakllantiradigan metod hisoblanadi. Biz quyida koʻrib chiqadigan algoritmlarni loyihalash texnikasiga asoslangan qadamlar boʻlib, qoʻyilgan masalani toʻgʻri va samarali yechib berish usullarini ochib beradi.

Qoʻyilgan masalaning turiga qarab algoritmlar ishlab chiqiladiki, bitta muammoning yechimi bir necha

usulda yechish mumkinki, qat'iy bir usulni belgilab olish xatodir. Lekin algoritmlarni ishlab chiqishda quyidagi bosqichlarga amal qilish berilgan masalani samarali hal qilishga olib keladi.

Amaliy jihatdan algoritmlarni loyihalashdan oldin berilgan loyihani to'liq tushunish muhimdir. Berilgan masalani to'liq tushunib olinganligiga ishonch hosil qilinib, keyin qaror qabul qilish kerak.

Muammoni to'liq tushunib yetgandan keyin, algoritm uchun mo'ljallangan hisoblash qurilmasini imkoniyatlarini ko'rib chiqish kerak. Biz kompyuter xotirasi yoki protsessor haqida o'ylashimiz kerakmi? Agar siz o'rganish bosqichida algoritm tuzayotgan bo'lsangiz bu haqda qayg'urmasangiz ham bo'ladi. Lekin, aniq bir masalalarni hal qilishga mo'ljallangan, katta ma'lumotlar bazasi bilan ishlaydigan algoritmlar haqida gapirsak, albatta buni inobatga olishimiz kerak. Masalan, biz ro'yxat elementlari bilan ishlaydigan algoritm yoqsak, berilgan ro'yxatning 20-indeksida turgan elementgacha chiziqli qidiruv (linary reserch) bilan tuzilgan algoritmda yigirma marta xotira va protsessor o'rtasida borib kelish jarayonini amalga oshiradi. Bu esa o'z navbatida aytish mumkinki, katta hajmli ro'yxat elementlarida indeks katta raqamli bo'lsa ancha ko'p resurs ya'ni protsessor va xotira xizmatini egallaydi.

Bugungi kunda biz sekin ishlashi bilan noliydigan kompyuterlarimiz protsessori chastotasi  $F=1.10\text{GHz}$  ya'ni bu kompyuter 1 daqiqada  $1.10 \cdot 10^9$  (milliard) amal bajaradi degani. Bu ham aslida yuqori ko'rsatkich bo'lsada bugunda bu tezlikdagi kompyuterlar ishlash samaradorligi past hisoblanadi. Chunki grafik imkoniyatlari yuqori va internet bilan bog'liq dasturlarda ishlashimiz tufayli bu parametrlar sifatli kompyuter xizmatini ko'rsatib bera olmasligi mumkin. Algoritmlar ishlash tezligini o'lchab beruvchi mezon bu Big O (Big Order complexity-katta buyurtma murakkabligi) hisoblanadi. Bu algoritmdagi eng minimal tezlikni o'lchaydi. O'lchov birligi opreatsiyalar soni hisoblanadi. Chiziqli qidiruv algoritm uchun  $O(n)$ , Binar qidiruv (Binary search)algoritmlarda  $O(\log_2 N)$  bilan o'lchanadi.

Keyingi asosiy qadam muammoni aniq hal qilish yoki taxminiy hal qilish o'rtasida bir qarorga kelishdir. Aniq yechimga ega algoritmlardan tashqari yana algoritmgaga yaqinlashi algoritmi deb ataluvchi loyihalar ham borki, masalan, kvadrat ildizlarni chiqarish, aniq integrallarni baholash kabilar shular jumlasidandir. Bu darajadagi masalalar ichki murakkabligi tufayli masalani yechimini sekinlashtirishi mumkin.

Ba'zi algoritmlar sodda ma'lumotlar bilan ishlasa, ba'zilar murakkab ma'lumotlar tuzilmasiga bog'liq va o'z navbatida zukkolikni va aqlni talab qiladi. Ma'lumotlar tuzilmasini tegishli tahlil qilib, algoritmlarni loyihalansa, sifati yaxshilanadi.

Eng asosiy qadamlardan biri bu-algoritmlarni loyihalash hisoblanadi. Algoritmlarni loyihalash bu (strategiya yoki paradigma ham deyiladi)turli sohalarida qo'llanilishi qo'llanilishi mumkin bo'lgan algoritmik muammolarni hal qilish umumiy yondashuvi hisoblanadi. Nima uchun bu texnikaga urg'u berishimiz kerak. Bu soha hali yangi masalalarni yechishda ko'rsatmalar beradi. Talaba bir qolipga tushib qolmasdan yangilik qilishga, uni amaliy sinab ko'rishga imkon beradi.

Algoritmlarni yaratgandan so'ng uni qaysi bir tarzda belgilab olishimiz kerakki, u bosqichma-bosqich bo'lsin hamda tushunarli. Odatda, ikkita usul juda keng tarqalgan: birinchidan, tabiiy tildan foydalanish. Bu juda ommalashgan bo'lsada ba'zi hollarda ya'ni boshqa tildagi foydalanuvchilar uchun bu tushunarsiz bo'lishi mumkin. Ikkinchi keng tarqalgan usul, pseudocode-ya'ni tabiiy til va zamonaviy dasturlash tilining uyg'unligidagi algoritmlar ifodalash usuli. Bu usul aniq bir qonun-qoidalarga bo'ysunmaydi va dasturchilar uchun tushunarli.

Hisoblash texnikasining rivojlanish bosqichlarida flowchart-blok-sxemalarda ifodalangan algoritmlar keng tarqalgan edi. Bular algoritmik qadamlarni geometrik shakllarda ifodalagan edi. Bu ifodalash usuli faqat oddiy algoritmlar uchun qulay bo'lib, murakkab algoritmlarni ochib berish qiyinchilik tug'dirardi. Bugungi kunda bu turdagi algoritmlarni deyarli uchratmaymiz.

Algoritmni ishlab chiqqandan keyin biz uni to'g'riligini isbotlashimiz kerak. Ya'ni cheklangan vaqtda, chekli qadamlar bilan to'g'ri natija berishini isbotlashimiz kerak. Ba'zi algoritmlarni isbotlash juda oson, ba'zilar juda murakkab bo'lishi mumkin. To'g'riligini isbotlashning eng keng tarqalgan usuli matematik induksiyadan foydalanish usulidir. Chunki algoritmlar iteratsiyasi bu kabi isbotlar uchun zarur ketma-ketlikni ta'minlaydi. Algoritm noto'g'ri ekanligini ko'rsangiz uning ma'lumotlar tuzilmasi yoki dizayn texnikasi bo'yicha qadamlarni qayta ko'rib chiqishingiz kerak.

Algoritmni to'g'riligini tekshirib ko'rgandan keyin eng asosiysi uning samaradorligi hisoblanadi. Vaqt samaradorligi algoritm qanchalik tez ishlashini ko'rsatsa, joy samaradorlik qancha qo'shimcha xotira olganligini ko'rsatadi. Algoritmlarning yana bir muhim jihati ularning oddiyligidadir. Chunki soddaroq algoritmlarni dasturlash oson bo'lib bu o'z navbatida dasturlash jarayonidagi xatolarni oldini oladi. Agar algoritm samaradorligi, soddaligi yoki umumiyliigi sizni qoniqtirmasa yuqorida aytib o'tilgan qadamlarni qayta ko'rib chiqishga to'g'ri

keladi.

Xulosa. Algoritmni loyihalashda yuqorida berib o'tilgan qadamlar ularning to'g'ri tuzilishini, sodda va tushunarligini hamda kompyuter resurslaridan samarali foydalanish bilan bir qatorda, uni dasturlash jarayonidagi xatolarni kamaytirib, tushunarli dastur tuzishga imkon yaratadi.

Foydalanilgan adabiyotlar:

1. Anany Levitin. Introduction to the design & analysis of algorithms. 2005
2. Anvar Narzullayev. <https://www.youtube.com/watch/algortimdarlari>
3. R.Robert Gajevski. Algorithms and programming flowcharts and flowgorithm.us.edu.pi
4. Kurt Mehlhom. Algorithms and programs. Resources.mpi-inf.mpg.de
5. Jurayeva N.O. Fundamentals of Organizing Students' Independent Work Using Mobile Applications. Child Studies in Asia-Pacific Context (CSAC). 2022, 12 (1); 255-266
6. Жўраева Н.О. Организация самостоятельного обучения по предмете «информационно-коммуникационные технологии». International Conference on New Scientific Methodologies (online-conferences). -327-331
7. Jo'rayeva N.O. Mobile Softwareanwendungen zur Organisation unabhängiger Bildung// Berlin Studies Transnational Journal of Science and Humanities. Vol. 2, Issue 1.5 (2022), – P. -661-664.
8. Jurayeva N.O. Specific aspects and principles of the method of organizing independent education of students. Actual problems of modern science, education and training". №8, Xorazm, 2022. – P. 23-27
9. Н.О.Жўраева. Таълим жараёнида мустақил ўқув фаолиятини ташкил этиш бўйича айрим кўрсатмалар. Таълим ва инновацион тадқиқотлар. №3, 2021 йил. -170-176 б
10. Sameer Punjal.Fibonacci Algorithms. Eskeype.github.io.2018