

## QIDIRISH ALGORITMLARI

## АЛГОРИТМЫ ПОИСКА

## SEARCH ALGORITHMS

**Abdullayev Shaxboz Solijon o‘g‘li***FarDU Axborot texnologiyalari kafedrası kata o‘qituvchisi*[shaxbozfardu2023@gmail.com](mailto:shaxbozfardu2023@gmail.com)ORCID ID [0000-0001-9382-732X](https://orcid.org/0000-0001-9382-732X)**Saminova Ruxshonaxon Muxtorjon qizi***Farg‘ona davlat universiteti Axborot tizimlari va texnologiyalari yo‘nalishi**I kurs talabasi*[saminovaruxshona551@gmail.com](mailto:saminovaruxshona551@gmail.com)

**Annotatsiya.** Ushbu maqolada qidirish algoritmlarining nazariy asoslari va ularning amaliy qo‘llanish xususiyatlari tahlil qilingan. Algoritmik yondashuv sifatida chiziqli qidirish, ikkilik qidirish, hash qidirish va qidiruv daraxtlari modellarining ishlash prinsiplari, vaqt va xotira murakkabliklari tahlili hamda ularning afzallik va kamchiliklari ko‘rib chiqilgan. Har bir algoritmnining samaradorligini ta‘minlovchi omillar, xususan ma‘lumotlar strukturasining o‘ziga xosligi va tartiblanganligi holatlari batafsil tahlil etilgan. Maqolada, shuningdek, qidirish algoritmlarini tanlash mezonlari va ular asosida qaysi holatda qaysi usul samaraliroq ekani asoslangan holda bayon etilgan. Tadqiqot natijalari qidirish algoritmlarini tanlashda amaliy tavsiyalar ishlab chiqish imkonini beradi hamda dasturiy ta‘minot, ma‘lumotlar bazasi boshqaruvi va sun‘iy intellekt sohalarida algoritmik samaradorlikni oshirishga xizmat qiladi.

**Аннотация.** В данной статье проанализированы теоретические основы алгоритмов поиска и особенности их практического применения. Рассмотрены принципы работы, временная и пространственная сложность, а также достоинства и недостатки основных моделей, таких как линейный поиск, бинарный поиск, хеширование и поисковые деревья. Подробно проанализированы факторы, влияющие на эффективность каждого

алгоритма, в частности структура и упорядоченность данных. Кроме того, в статье определены критерии выбора алгоритмов поиска и обосновано, в каких условиях тот или иной метод является наиболее эффективным. Результаты исследования предоставляют практические рекомендации по выбору алгоритмов поиска и способствуют повышению алгоритмической эффективности в таких областях, как разработка программного обеспечения, управление базами данных и искусственный интеллект.

**Annotation.** This article analyzes the theoretical foundations of search algorithms and their practical application features. The working principles, time and space complexity, as well as the advantages and disadvantages of key models such as linear search, binary search, hashing, and search trees are discussed. The factors influencing the efficiency of each algorithm, particularly the structure and ordering of data, are examined in detail. The article also outlines the criteria for selecting search algorithms and explains which methods are more efficient under specific conditions. The results of the study provide practical recommendations for the selection of search algorithms and contribute to enhancing algorithmic efficiency in fields such as software development, database management, and artificial intelligence.

**Kalit so‘zlar:** Qidirish algoritmlari, chiziqli qidirish, ikkilik qidirish, hash qidirish, qidiruv daraxtlari, vaqt murakkabligi, ma'lumotlar strukturalari, algoritmik samaradorlik, tartiblangan ma'lumotlar, sun'iy intellekt, dasturlash, ma'lumotlar bazasi boshqaruvi.

**Ключевые слова:** Алгоритмы поиска, линейный поиск, бинарный поиск, хеширование, поисковые деревья, временная сложность, структуры данных, алгоритмическая эффективность, упорядоченные данные, искусственный интеллект, программирование, управление базами данных.

**Keywords:** Search algorithms, linear search, binary search, hashing, search trees, time complexity, data structures, algorithmic efficiency, sorted data, artificial intelligence, programming, database management.

## Kirish

Qidirish algoritmlari — bu ma'lumotlar bazasida yoki ma'lumotlar strukturasida kerakli ma'lumotni topish uchun ishlatiladigan metodlar to'plamidir. Ular kompyuter

fanlari va dasturlashda juda muhim o‘rin tutadi, chunki samarali qidirish tizimlari ma'lumotlar bilan ishlashda vaqtni va resurslarni tejashga yordam beradi. Qidirish algoritmlari turli tizimlarda, shu jumladan ma'lumotlar bazalarida, fayl tizimlarida, tarmoq arxitekturalarida va sun'iy intellekt sohalarida keng qo'llaniladi. Ular, shuningdek, mobil ilovalar, veb qidiruv tizimlari va hatto raqamli asarlarda ham muhim ahamiyatga ega. Qidirish algoritmlari odatda ma'lumotlar strukturasi qanday tashkil etilganiga qarab turli xil bo'ladi. Masalan, agar ma'lumotlar ro'yxat tarzida berilgan bo'lsa, linear qidirish algoritmi samarali bo'ladi. Agar ma'lumotlar tartiblangan bo'lsa, binary qidirish algoritmi yaxshiroq ishlaydi. Hashing esa ma'lumotlarga kirishning eng tezkor usullaridan biri hisoblanadi. Graf va daraxt kabi murakkab strukturalarda esa o'ziga xos qidirish metodlari, masalan, DFS (Depth-First Search) va BFS (Breadth-First Search) qo'llaniladi. Qidirish algoritmlarining samaradorligi asosan ikki omilga bog'liq: ularning ishlash vaqti va xotira sarfi. Agar algoritmi ma'lumotlarni juda tez topa olsa, lekin xotira sarfi yuqori bo'lsa, bu tizimning umumiy samaradorligini kamaytirishi mumkin. Shuning uchun qidirish algoritmlarini tanlashda vaqt va resurslar jihatidan eng samarali variantni tanlash muhimdir.

Ushbu maqolada qidirish algoritmlarining turli turlari, ularning ishlash prinsiplari va qo'llanilish sohaları to'g'risida batafsil ma'lumot beriladi. Qidirish algoritmlarining to'g'ri tanlanishi va optimallashtirilishi nafaqat tizim samaradorligini oshiradi, balki katta hajmdagi ma'lumotlar bilan ishlashda muhim rol o'ynaydi. Maqola davomida linear, binary qidirish, hashing, va graf qidirish algoritmlari kabi asosiy turlarini va ularning afzallik hamda kamchiliklarini ko'rib chiqamiz.

### ➤ **Linear qidirish**

Linear qidirish algoritmi eng oddiy va tushunarli qidirish metodidir. U barcha elementlarni birma-bir tekshirib chiqadi, shuning uchun uning vaqti  $O(n)$  bo'ladi, bu yerda  $n$ -ro'yxatdagi elementlar soni.

#### **Ishlash tartibi:**

- Elementni ro'yxatning har bir qismiga takshirib chiqadi.
- Agar element topilsa, algoritmi to'xtaydi, aks holda davom etadi.

#### **Afzalliklari:**

- Oddiy va amalga oshirish oson.

- Ma'lumotlar tartiblanmagan bo'lsa ham ishlaydi.

**Kamchiliklari:**

- Tezlik jihatidan samarali emas, katta ro'yxatlar bilan ishlaganda juda sekin bo'lishi mumkin.

➤ **Binary Qidirish**

Binary qidirish faqat tartiblangan ro'yxatlar bilan ishlaydi va tezroq natijalarga erishish uchun ma'lumotlar orasida qidirishni yarmiga bo'lib, har safar yangi qismda izlashni davom ettiradi. Bu algoritmnining samaradorligi  $O(\log n)$  bo'ladi, bunda  $n$  — ro'yxatdagi elementlar soni.

**Ishlash tartibi:**

- Ro'yxatdagi o'rtacha elementni toping.
- Agar qidirilayotgan element bu o'rtadagi elementdan kichik bo'lsa, faqat chap qismda qidiring, aks holda o'ng qismda.
- Bu jarayonni ro'yxatning qolgan qismlariga nisbatan takrorlang.

**Afzalliklari:**

- Katta ro'yxatlarda juda samarali.
- Agar ro'yxat tartiblangan bo'lsa, eng tezkor algoritmi.

**Kamchiliklari:**

- Faqat tartiblangan ma'lumotlar bilan ishlaydi.
- Kichik ro'yxatlar uchun samarali emas.

➤ **Hashing**

Hashing — bu maxsus "hash" funktsiyasini ishlatib, elementni ma'lum bir manzilga joylashtirishni o'z ichiga oladi. Bu metod orqali qidirish, qo'shish va o'chirish operatsiyalari tez amalga oshiriladi. Hashingning samaradorligi odatda  $O(1)$ .

**Ishlash tartibi:**

- Har bir element uchun hash funktsiyasi yordamida kalitlar yaratiladi.
- Kiritilgan kalit bo'yicha to'g'ri manzilga yoki joyga o'tish amalga oshiriladi.

**Afzalliklari:**

- Tez qidirish va o'chirish imkoniyati.
- Katta hajmdagi ma'lumotlar uchun samarali.

**Kamchiliklari:**

- Kollisionlar (hash kalitlarining takrorlanishi) bilan bog'liq muammolar.
- Dasturdan foydalanishdan oldin ma'lum bir sozlashlarni talab qiladi.

**➤ Grafda qidirish**

Grafda qidirish algoritmlari (DFS, BFS) grafdagi tugunlar (nodlar) orasida ma'lumot izlash uchun ishlatiladi. Bu metodlar ayniqsa grafning bog'liqligi va turli tizimlar orasida yo'l topish uchun mos keladi.

DFS (Depth-First Search): Qidirish tugunlardan chuqurroq bo'yicha davom etadi. Foydalanuvchi birinchi bo'lib imkoniyatlarni to'liq o'rganishga harakat qiladi.

BFS (Breadth-First Search): Qidirish kengroq bo'yicha amalga oshiriladi, ya'ni barcha qo'shni tugunlar birinchi bo'lib tekshiriladi.

**Xulosa**

Xulosa qilib aytganda, qidirish algoritmlari ma'lumotlar bilan ishlash jarayonining ajralmas va muhim qismi bo'lib, ular yordamida ma'lumotlarga tezkor va samarali tarzda kirish mumkin bo'ladi. Har bir algoritmnining o'ziga xos ustunlik va cheklovlari mavjud bo'lib, ularni tanlashda ma'lumotlar tuzilmasi va vazifalarning xususiyatlari inobatga olinishi zarur. Linear qidirish oddiy va universal bo'lsa, binary qidirish tartiblangan ma'lumotlar uchun yuqori samaradorlikni ta'minlaydi, hashing esa katta hajmdagi ma'lumotlar bilan ishlashda beqiyos natijalar beradi.

Graf asosidagi qidirish algoritmlari esa murakkab tizimlar va bog'lanishlar orasida ma'lumot izlashda keng qo'llaniladi. Bugungi kunda qidirish algoritmlarining roli nafaqat klassik dasturlashda, balki sun'iy intellekt, ma'lumotlar analitikasi va avtomatlashtirilgan tizimlar rivojida ham juda katta. Shu sababli, qidirish algoritmlarini chuqur o'rganish va ulardan to'g'ri foydalanish har bir zamonaviy mutaxassis uchun muhim bilim va amaliy ko'nikma sanaladi.

**Foydalanilgan adabiyotlar**

1. Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C. *Introduction to Algorithms* (3rd Edition), The MIT Press, 2009.
2. Knuth, D. E.  
The Art of Computer Programming, Volume 1-4, Addison-Wesley, 1997.
3. Sedgewick, R., Wayne, K.

Algorithms (4th Edition), Addison-Wesley, 2011.

4. Baxtiyor Saidov, Temur G‘ulomov.

Algoritmlash va dasturlash asoslari, TATU nashriyoti, Toshkent, 2020.

5. S. Obidov, M. Qodirov.

Dasturlash asoslari va algoritmik tafakkur, TDYU nashriyoti, 2021.

6. Tim Roughgarden.

Algorithms Illuminated (Vol. 1–4), Soundlik LLC, 2018–2020.

7. Ullman, J. D., Hopcroft, J. E., Motwani, R.

Automata Theory, Languages and Computation, Pearson, 2006.