

KATTA HAJMDAGI SONLAR BILAN ISHLASH YOKI XOTIRA TEJASHDA BITLI ALGORITMLAR VA ULARNING METODLARIDAN FOYDALANISH

Farmonov Sherzodbek Rahmonjonovich

Farg'ona davlat universiteti amaliy matematika va
informatika kafedrası katta o'qituvchisi.

farmonovsh@gmail.com

Farg'ona Davlat Universiteti Amaliy matematika
yo'nalishi 23.10-guruh talabasi Soraxon Yorbekova

gulsoraxonyorbekova@gmail.com

Annotatsiya: Ushbu maqolada bitli algoritmlar tahlil qilinadi va ularning samaradorligi o'rganiladi. Bitli algoritmlarning asosi, ulardan amaliy masalalarda (masalan, katta hajmdagi sonlar bilan ishlash yoki xotira tejash) foydalanish imkoniyatlari ko'rib chiqiladi. Bitli operatsiyalarning afzalliklari va C# dasturlash tilida ularni amalga oshirish misollari taqdim etiladi.

Kalit so'zlar: Bitli algoritmlar, bit operatsiyalari, dasturlash, xotira optimallashtirish, samaradorlik.

Annontation: This article analyzes bit algorithms and evaluates their efficiency. The basics of bit algorithms and their application in practical problems (e.g., working with large numbers or saving memory) are examined. The advantages of bit operations and examples of their implementation in the C# programming language are presented.

Keywords: Bit algorithms, bit operations, programming, memory optimization, efficiency.

Аннотация: В данной статье анализируются битовые алгоритмы и оценивается их эффективность. Рассматриваются основы битовых

алгоритмов и их применение в практических задачах (например, работа с большими числами или экономия памяти). Приведены преимущества битовых операций и примеры их реализации на языке программирования C#.

Ключевые слова: Битовые алгоритмы, битовые операции, программирование, оптимизация памяти, эффективность.

Bitli algoritmlar — bu kompyuterda ma'lumotlarni tahlil qilish, qayta ishlash va saqlash jarayonlarida bit operatsiyalaridan foydalanadigan maxsus algoritmlardir. Bit operatsiyalari asosida ishlovchi algoritmlar xotirani tejash va hisoblash tezligini oshirish uchun samarali vositadir. Ushbu maqolada bitli algoritmlarning ishlash tamoyillari, ular yordamida masalalarni yechish usullari va dasturiy ta'minotdagi tatbiqi tahlil qilinadi. Bit-bu raqamli dunyoning eng kichik ma'lumot birligi bo'lib, u faqat ikkita qiymatga ega bo'lishi mumkin: 0 yoki 1. Bitli algoritmlar bit operatsiyalaridan (masalan, AND, OR, XOR, NOT) foydalanadi. Bu operatsiyalar kompyuter protsessorida juda tez ishlaydi, chunki ular past darajadagi apparat darajasida qo'llaniladi. Algoritmlar, masalan, ketma-ketliklar bilan ishlash, yig'indilarni hisoblash yoki eng kichik/maksimal qiymatlarni topish uchun foydalaniladi. Bitli algoritmlar — bu ma'lumotlarni eng past darajada, ya'ni kompyuterda ma'lumotlar ishlov beriladigan asosiy birlik, bitlar orqali manipulyatsiya qiladigan algoritmlar. Ular kompyuterning o'zi foydalanadigan ikkilik (0 va 1) tizimida ishlaydi va odatda juda tez va samarali hisoblanadi.

Bitli algoritmlar (bitwise algorithms) — bu ma'lum bir vazifalarni bajarishda ma'lumotlarni bitlar darajasida ishlashga asoslangan algoritmlar bo'lib, ular kompyuterda tez va samarali ishlashni ta'minlaydi. Ular odatda integer (butun son) tipidagi qiymatlarni ishlatadi va ba'zan eng past darajadagi operatsiyalarni qo'llab quvvatlaydi. Amaliyotda bitli operatsiyalar bir qator vazifalarni hal qilish uchun ishlatiladi.

Masalan:

1. Katta hajmdagi massivlarni siqish va ularda ma'lumotlarni saqlash.

2. Kriptografiya algoritmlarini amalga oshirish.

3. Matematika va statistika sohasida hisoblashlar uchun optimal yondashuv. Bitli algoritmlar ko'pincha quyidagi amallarni bajarish uchun ishlatiladi:

Bitlarni manipulyatsiya qilish: Bitlarni o'zgartirish (0 ni 1 ga yoki 1 ni 0 ga aylantirish).

Ma'lum pozitsiyadagi bitni aniqlash yoki o'zgartirish. Butun sonlarning bit bo'yicha taqqoslanishi yoki ishlov berilishi. Ma'lumotlarni ixchamlash va shifrlash: Bitli algoritmlar ma'lumotlarni kompresslashda va shifrlashda qo'llaniladi, chunki ular katta hajmdagi ma'lumotlarni samarali kodlash imkonini beradi.

Jadval yoki flaglar orqali belgilarni kuzatish: Bitlar to'plami ko'pincha ko'p sonli belgilarni (flaglar) kuzatish uchun ishlatiladi. Matematik hisob-kitoblar: Ko'paytirish, bo'lish yoki qoldiqni hisoblash kabi ba'zi matematik operatsiyalarni optimallashtirishda bitli amallar qo'llaniladi.

Quyida bitli algoritmlarning asosiy qo'llanilish sohalari keltirilgan:

1. Kriptografiya va ma'lumotlarni himoyalash:

Hash funksiyalari: Bitli operatsiyalar kriptografik hash funksiyalarni amalga oshirishda ishlatiladi, masalan, SHA va MD5 algoritmlari.

Shifrlash algoritmlari: AES, DES kabi algoritmlar bitli operatsiyalarga asoslanadi.

Kalit yaratish va boshqarish: Bit manipulyatsiyasi kalit generatsiyasini samarali amalga oshirish imkonini beradi.

2. Kompyuter grafikasi va o'yin dasturlari:

Piksel operatsiyalari: Rangni o'zgartirish, aralashtirish yoki filtrlash kabi operatsiyalar bitli manipulyatsiyaga asoslangan.

Matritsa optimizatsiyasi: Grafik tasvirlarni tezroq qayta ishlash uchun bitli algoritmlar ishlatiladi.

3. Signal va ma'lumotlarni siqish:

Ma'lumotlarni kodlash va dekodlash: Huffman kodlash, Run-Length Encoding (RLE) kabi siqish algoritmlarida bitli operatsiyalar qo'llaniladi.

Audio va video siqish: MP3, MPEG kabi formatlarda ishlov berish jarayonlari bitlar darajasida amalga oshiriladi.

4. Mashina o'qitishi va sun'iy intellekt:

Tezkor o'xshashliklarni aniqlash: Mahsulotlarni qidirish yoki tasvirni solishtirishda min-hash yoki bitli distance algoritmlari ishlatiladi.

Bool algebra: Neyron tarmoqlar uchun engil va samarali hisob-kitoblar.

5. Operatsion tizimlar va apparat ishlov berish:

Fayllarni boshqarish: Operatsion tizimlarda fayl flaglarini o'rnatish yoki o'chirish uchun ishlatiladi.

Keshlash: Kesh xotira strategiyasini amalga oshirishda bit operatsiyalari keng qo'llaniladi.

6. Tarmoq va protokollar:

Xabarlarni tekshirish: Parity bit va CRC (Cyclic Redundancy Check) kabi xatolarni aniqlash va tuzatish algoritmlari.

IP-manzillarni boshqarish: IP ni maskalash, subnet hisoblash bitli operatsiyalar yordamida amalga oshiriladi.

7. Matematik va statistik dasturlar:

Tez hisobotlar: Ko'paytirish, bo'lish va boshqa arifmetik operatsiyalarni optimallashtirish.

O'yin nazariyalari: Kombinatorika va o'yinlar strategiyasida bitli xaritalar qo'llaniladi.

Masala: Sonlar to'plamidagi juft bitlarni hisoblash

Berilgan to'plamdagi sonlarning ikkilik ko'rinishida **1** bo'lgan bitlarning jami sonini aniqlash kerak.

Masala: Agar to'plamda sonlar 5 (0101), 7 (0111) va 10 (1010) bo'lsa,

- **5** uchun 1-bitlar soni: 2

- 7 uchun 1-bitlar soni: 3
- 10 uchun 1-bitlar soni: 2

Jami: $2 + 3 + 2 = 7$

C# kodi:

```
using System;
class BitliMasala
{
    static void Main(string[] args)
    {
        // Sonlar to'plami
        int[] numbers = { 5, 7, 10 };
        // Jami 1-bitlar soni
        int totalBits = 0;
        Console.WriteLine("Sonlar va ularning 1-bitlari soni:");
        foreach (int number in numbers)
        {
            int count = CountSetBits(number);
            totalBits += count;
            Console.WriteLine($"Son: {number}, 1-bitlar soni: {count}");
        }
        Console.WriteLine($"
Jami 1-bitlar soni: {totalBits}");
    }
    // Ikkilik ko'rinishida 1-bitlarni hisoblash funksiyasi
    static int CountSetBits(int num)
    {
        int count = 0;
        while (num > 0)
        {
```

```
        count += num & 1; // Oxirgi bitni tekshirish
        num >>= 1;      // Bitlarni bir joyga siljitish
    }
    return count;
}
}
```

Kodning ishlash izohi

1. **CountSetBits** funksiyasi: Sonning ikkilik ko‘rinishida 1 bo‘lgan bitlarni hisoblaydi. Bu jarayon bitlarni birma-bir o‘ngga surib, oxirgi bitni tekshirish orqali amalga oshiriladi.
2. **Jami hisoblash:** Har bir son uchun 1-bitlar soni hisoblanib, yakuniy natijaga qo‘shiladi.
3. **Natijalarni chiqarish:** Har bir son va ularning 1-bitlari soni ekranga chiqariladi.

Dastur natijasi:

Agar sonlar to‘plami {5, 7, 10} bo‘lsa, dastur quyidagicha natija beradi:

Sonlar va ularning 1-bitlari soni:

Son: 5, 1-bitlar soni: 2

Son: 7, 1-bitlar soni: 3

Son: 10, 1-bitlar soni: 2

Jami 1-bitlar soni: 7

Bitli algoritmlar yuqori tezlikda ishlashi tufayli ko‘plab amaliy sohalarda, jumladan, dasturlash, kriptografiya va tarmoq xavfsizligida muvaffaqiyatli qo‘llaniladi. Ushbu maqolada bitli algoritmlarning nazariy va amaliy jihatlari tahlil qilindi va ularning imkoniyatlari ochib berildi.

Foydalanilgan adabiyotlar:

1. Marcin Jamro. *C# Data Structures and Algorithms*. Second Edition. Published by Packt Publishing Ltd., in Birmingham, UK. 2024. – 349 p.

2. Дж.Эриксон. Алгоритмы.: – М.: " ДМК Пресс ", 2023. – 528 с.
3. Hemant Jain. Data Structures & Algorithms using Kotlin. Second Edition. in India. 2022. – 572 p.
4. Н. А. Тюкачев, В. Г. Хлебостроев. С#. Алгоритмы и структуры данных: учебное пособие для СПО. – СПб.: Лань, 2021. – 232 с.
5. Mykel J. Kochenderfer. Tim A. Wheeler. Algorithms for Optimization. Published by The MIT Press., in London, England. 2019. – 500 p.
6. Рафгарден Тим. Совершенный алгоритм. Графовые алгоритмы и структуры данных. – СПб.: Питер, 2019. - 256 с.
7. [Ахо Альфред В., Ульман Джеффри Д., Хопкрофт Джон Э.](#) Структуры данных и алгоритмы. – М.: [Вильямс](#), 2018. – 400 с.
8. Дж.Хайнеман, Г.Поллис, С.Стэнли. Алгоритмы. Справочник с примерами на С, С++, Java и Python, 2-е изд.: Пер. с англ. — СПб.: ООО "Альфа-книга", 2017. — 432 с.
9. Raxmonjonovich, F. S. (2024). MA'LUMOTLARNI SIQISHDA BITLI ALGORITMLARDAN FOYDALANISH. Modern education and development, 15(5), 320-328.
10. Raxmonjonovich, F. S. (2024). AXBOROTLARNI SHIFRLASHDA MATEMATIK ALGORITMLARDAN FOYDALANISH. Modern education and development, 15(5), 338-344.
11. Raxmonjonovich, F. S. (2024). BIR SHAHARDAN BOSHQASIGA YUK YETKAZIB BERISHDA ENG OPTIMAL VA KAM XARAJAT SARFLANADIGAN YO'LNI TOPISHDA BELLMAN-FORD ALGORITMIDAN FOYDALANISH. Ta'lim innovatsiyasi va integratsiyasi, 34(2), 72-78.
12. Raxmonjonovich, F. S. (2024). KOMPYUTER TARMOQLARI SOHASIDA BITLI ALGORITMLAR. Modern education and development, 15(4), 50-59.