

BITLI OPERATSIYALARNI SAMARALI OPTIMALLASHTIRISH USULLARI

Farmonov Sherzodbek Rahmonovich

*Farg'ona davlat universiteti amaliy matematika va
informatika kafedrası katta o'qituvchisi
farmonovsh@gmail.com*

Meliqo'ziyeva Gulsanam Po'latjon qizi

*Farg'ona davlat universiteti 2-kurs talabasi
gulsanammeliqoziyeva18@gmail.com*

Hamidov Hayotbek Oqilbek o'g'li

*Farg'ona davlat universiteti 2-kurs talabasi
hamidovhayotbek76@gmail.com*

Annotatsiya. Ushbu maqolada bitli algoritmlar va ularning zamonaviy hisoblash texnologiyalaridagi o'rni tahlil qilinadi. Maqolada ushbu algoritmlarning ishlash tamoyillari, ularni qo'llash sohalari hamda texnik imkoniyatlari ko'rib chiqiladi. Bitli algoritmlarning amaliyotdagi afzalliklari va cheklovlari muhokama qilinib, samarali algoritmlar yaratish bo'yicha tavsiyalar beriladi. Tadqiqot natijalari algoritmlar bilan ishlashda innovatsion yondashuvlarni rivojlantirishga xizmat qiladi.

Kalit so'zlar: bitli algoritmlar, ma'lumotlarni qayta ishlash, bit operatsiyalari, algoritmik samaradorlik, dasturlash optimizatsiyasi, kompyuter arxitekturası, kriptografiya algoritmlari, ma'lumotlarning siqilishi, tezkor hisoblash texnikasi, algoritmik murakkablik.

Annotation. This article analyzes bitwise algorithms and their role in modern computing technologies. It examines the principles of operation of these algorithms, their areas of application, and technical capabilities. The practical advantages and limitations of bitwise algorithms are discussed, and recommendations for creating efficient algorithms are provided. The research findings contribute to the development of innovative approaches to working with algorithms.

Keywords: bitwise algorithms, data processing, bit operations, algorithmic efficiency, programming optimization, computer architecture, cryptographic algorithms, data compression, high-performance computing techniques, algorithmic complexity.

Аннотация. В данной статье анализируются побитовые алгоритмы и их роль в современных вычислительных технологиях. Рассматриваются принципы работы этих алгоритмов, их области применения и технические возможности. Обсуждаются практические преимущества и ограничения побитовых алгоритмов, а также даются рекомендации по созданию эффективных

алгоритмов. Результаты исследования способствуют разработке инновационных подходов к работе с алгоритмами.

Ключевые слова: побитовые алгоритмы, обработка данных, битовые операции, алгоритмическая эффективность, оптимизация программирования, компьютерная архитектура, криптографические алгоритмы, сжатие данных, методы высокопроизводительных вычислений, алгоритмическая сложность.

Zamonaviy kompyuter texnologiyalari ma'lumotlarni tezkor va samarali qayta ishlashni talab qiladi. Ushbu jarayonda algoritmlarning tuzilishi va ishlash samaradorligi muhim ahamiyat kasb etadi. Bitli algoritmlar – ma'lumotlarni eng kichik birlikda ya'ni bit darajasida qayta ishlashga asoslangan algoritmlar bo'lib, ular tezlik va samaradorlik jihatidan ko'plab hisoblash masalalarini yechishda muhim rol o'ynaydi. Bit operatsiyalari zamonaviy dasturlash tillarida keng qo'llanilib, arifmetik hisoblash, ma'lumotlarni siqish, kriptografiya va boshqa sohalarda yuqori samaradorlikni ta'minlaydi.

Algoritmlarning ishlash jarayoni: Bitli algoritmlar ma'lumotlarni eng kichik birlik ya'ni bit darajasida qayta ishlaydi. Bu algoritmlar asosiy amallarni bajarish uchun bit operatsiyalaridan foydalanadi.

Ma'lumotlarni bitli ko'rinishga o'tkazilishi: Ma'lumotlar ikki tomonlama sanoq tizimida (binary) ifodalanadi. Bu ko'rinishda har bir qiymat bitlar ketma-ketligi sifatida saqlanadi va qayta ishlanadi.

Bit operatsiyalari bajarilishi: Bitli algoritmlar ma'lumotlarni qayta ishlashda quyidagi operatsiyalardan foydalaniladi

AND (va) – har ikkala bit qiymati 1 bo'lsa, natija 1 bo'ladi.

OR (yoki) – kamida bitta bir qiymati 1 bo'lsa, natija 1 bo'ladi.

XOR (eksklyuziv yoki) – faqat ikkita bit qiymati turlicha bo'lsa, natija 1 bo'ladi.

NOT (inkor) – bit qiymatini teskari qiladi.

Bit operatsiyalari oddiy arifmetik operatsiyalarga nisbatan kamroq vaqt va resurs talab qiladi. Shu sababli, katta hajmdagi hisoblashlarni tezkor bajarishda bitli algoritmlar samarali hisoblanadi. Masalan, arifmetik hisoblashlarni (ko'paytirish, bo'lish) siljitish orqali amalga oshirish mumkin. Bitli algoritmlar yordamida yechim topilgach, natijalar qayta umumlashtiriladi va boshqa formatga o'tkaziladi. Bu jarayon ko'pincha ilingan ma'lumotlarning moslashuvchanligiga bog'liq.

Amaliy qo'llanish: Bitli algoritmlar kriptografiya, ma'lumotlarni siqish, qidiruv algoritmlari, massivlarni saralashda qo'llaniladi. Ular tizim samaradorligini oshirishda va katta hajmdagi ma'lumotlarni qayta ishlashda muhim ahamiyatga ega.

Bitli algoritmlarning matematik tavsifi: Bitli algoritmlar matematik jihatdan ma'lumotlarni ikki tomonlama sanoq tizimida (binary system) ifodalash va bitlarga

asoslangan operatsiyalar orqali qayta ishlashni bildiradi. Ushbu algoritmlar bit operatsiyalari yordamida tezkor hisoblashlarni amalga oshiradi.

1. Chapga siljitish (Left Shift):

$$C = A \ll n$$

Matematik jihatdan bu A-ni 2^n ga ko'paytirishga teng: $C = A * 2^n$

2. O'ngga siljitish (Right Shift):

$$C = A \gg n$$

Matematik jihatdan bu A-ni 2^n ga bo'lishga teng: $C = A / 2^n$

Matematik modellash: Bitli algoritmlar turli murakkab masalalar uchun quyidagicha matematik modellashda ishlatiladi:

*Kriptografiya: Shifrlashda bitli XOR operatsiyasi keng qo'llaniladi

*Ma'lumotlarni siqsh: Takrorlanadigan bit ketma-ketliklarini tahlil qilish va qayta yozish orqali hajm kamaytiriladi.

*Arifmetik hisoblar: Ko'paytirish va bo'lish operatsiyalari quyidagicha amalga oshiriladi:

$$A * B = (A \ll k) \text{ or } (B \ll m).$$

Bitli algoritmlar matematik asosga ega bo'lib, samarali hisoblash va ma'lumotlarni qayta ishlash uchun qulay vosita.

Algoritmning natijalari va samaradorligi: Bitli algoritmlar ma'lumotlarni bit darajasida qayta ishlash orqali quyidagi natijalarga erishishga imkon beradi:

- Tezlik. Bitli operatsiyalar arifmetik operatsiyalarga qaraganda tezroq bajariladi. Masalan, ko'paytirish yoki bo'lish o'rniga siljitish amallarini ishlatish vaqtni sezilarli darajada tejaydi.

Masalan, $A * 2^n$ ni hisoblash uchun $A \ll n$ siljitish amalga oshiriladi, bu ko'paytirishga nisbatan samaraliroqdir.

- Hajmni optimallashtirish. Ma'lumotlarni siqish algoritmlarida takrorlanuvchi bit ketma-ketliklarini qayta yozish orqali ma'lumot hajmini kamaytirish mumkin. Bu, xususan, fayllarni siqish formatlarida (ZIP, PNG) keng qo'llaniladi.

- Kriptografiya. Bitli algoritmlar yuqori darajadagi xavfsizlikni ta'minlaydi. XOR operatsiyasi, masalan, ma'lumotlarni shifrlashda ishlatiladi, chunki u reversiv va oddiy hisoblanadi/

- Qo'llash sohalari.

Ma'lumotlarni indekslash va qidirish (Masalan, Bloom filter)

Grafiklarni qayta ishlash

Protsessorning past darajadagi optimizatsiyasi.

Samaradorlik:

1. Murakkablik darajasi. Bitli operatsiyalar odatda $O(1)$ vaqt murakkabligiga ega, chunki ular to'g'ridan – to'g'ri apparat darajasida amalga oshiriladi. Bu boshqa algoritmlarga qaraganda ulkan tezlikni ta'minlaydi.

2. Kam resurs talabi

Xotira – bitli algoritmlar minimal xotira sarflaydi, chunki ular ma'lumotlarni eng kichik birliklarda ishlaydi.

Protsessor yuklanishi – bit operatsiyalari kamroq protsessor sikllarini talab qiladi, bu esa quvvat sarfini kamaytiradi.

3. Moslanuvchanlik. Bitli algoritmlar ko'plab sohalarda qo'llanilishi mumkin va ular turli xil texnologiyalarga moslashadi:

Mobil qurilma uchun optimallashtirilgan hisoblashlar

Tizim darajasida ishlov berish

4. Amaliy natijalar

CRC – tarmoqdagi ma'lumotlar yaxlitligini tekshirish uchun bitli operatsiyalarni qo'llaydi.

Ma'lumotlarni kodlash va dekodlash – JPEG va MP3 kabi formatlarda bitli algoritmlar ishlatiladi.

Kelajakda rivojlantirish imkoniyatlari: Bitli algoritmlar bugungi kunda ko'plab sohalarda qo'llanilsada uni takomillashtirish uchun ba'zi usullar mavjud.

1. Yuqori samarali apparat integratsiyasi.

GPU va TPU optimizatsiyasi: Bitli algoritmlarni grafik protsessorlar (GPU) va tensor protsessor birliklari (TPU) bilan integratsiyalash orqali ma'lumotlarni qayta ishlash tezligi oshirilishi mumkin.

Kvant hisoblash: Kelajakda kvant kompyuterlari yordamida bitli algoritmlarni kvant bitlari darajasida qo'llab, yanada kuchli va samarali texnologiyalarni yaratish mumkin.

2. Sun'iy intellekt bilan integratsiyasi

Optimallashtirilgan o'rganish: Sun'iy intellekt va mashinaviy o'rganish algoritmlari yordamida bitli algoritmlarni turli ma'lumotlarga mos ravishda dinamik optimallashtirish mumkin.

Ko'p soxalarga moslashtirish: AI modellarida bitli algoritmlardan foydalanib, kata hajmdagi ma'lumotlarni tezkor qayta ishlash uchun yangi usullar yaratilishi mumkin.

3. Xavfsizlik va kriptografiya qo'llanishi

Post - quantum kriptografiyasi: Kvant hisoblash tahdidlariga qarshi yangi bitli algoritmlar asosida xavfsizlik kriptografik tizilar ishlab chiqish.

Ma'lumotlarni himoyalash: Ma'lumotlarni shifrlash va xakkerlikdan himoya qilish uchun bitli operatsiyalarga asoslangan murakkab mexanizmlarni joriy etish .

Muammo:

Tasodifiy sonlarni generatsiya qilishda ba'zi algoritmlar bit darajasida ishlaydi, bu aniqlikni oshirish va resurslarni tejash uchun muhim. Ushbu masalada Linear Feedback Shift Register (LFSR) algoritmini tahlil qilamiz. LFSR tasodifiy bit ketma-ketliklarini yaratishda ishlatiladi.

Masala Tavsifi

1. **Algoritm:** Linear Feedback Shift Register (LFSR) orqali tasodifiy bitlar yaratish.
2. **Aniqlik:** Generatsiya qilingan bitlarning haqiqatan tasodifiy ekanligini baholash.
3. **Unumdorlik:** Belgilangan bitlarni yaratish uchun sarflangan vaqtni o'lchash.

Linear Feedback Shift Register Algoritmining Asosiy Tamoyillari

LFSR algoritmi oddiy qadamlar orqali bitli tasodifiy sonlarni hosil qiladi:

1. **Boshlang'ich qiymat:** Birlamchi holatda registr tasodifiy bitlar bilan to'ldiriladi.
2. **Bitlarni qayta hisoblash:** XOR operatsiyasi orqali yangi bit hosil qilinadi.
3. **Shift qilish:** Registrdagi bitlar chapga yoki o'ngga siljiydi.

C# KODI: Linear Feedback Shift Register (LFSR)

```
using System;

class Program
{
    static void Main()
    {
        // LFSR boshlang'ich qiymatlari
        int seed = 0b10100101; // 8-bitli boshlang'ich qiymat
        int taps = 0b00000101; // XOR uchun foydalaniladigan tugunlar
        int register = seed; // Registr boshlang'ich qiymati
        int n = 16; // Generatsiya qilinadigan bitlar soni

        Console.WriteLine("Tasodifiy bitlar (LFSR):");
        for (int i = 0; i < n; i++)
        {
            // XOR operatsiyasi orqali yangi bitni hosil qilish
```

```

int newBit = (register & taps) % 2;

// Registrni chapga siljitish va yangi bitni qo'shish
register = (register >> 1) | (newBit << 7);

// Registrning oxirgi bitini tasodifiy bit sifatida ko'rsatish
Console.WriteLine(newBit);
}

Console.WriteLine();
}

```

Kodni Tushuntirish

1. **Seed:** Boshlang'ich qiymat sifatida 8-bitli son ishlatiladi (seed).
2. **Taps:** XOR operatsiyasi uchun bitlar qaysi joylarda ishlatilishini belgilaydi.
3. **Shift:** Registrni chapga siljitib, yangi bitni oxirgi o'ringa qo'shadi.
4. **Natija:** Har bir qadamda hosil qilingan yangi bit tasodifiy bit sifatida qaytariladi.

Bitli algoritmlar zamonaviy hisoblash texnologiyalarida samaradorlikni oshirish va resurslardan optimal foydalanish imkonini beruvchi muhim vositalardan biri sifatida e'tirof etiladi. Ularning qo'llanilish sohasi sun'iy intellekt, kriptografiya, IoT tizimlari, ma'lumotlarni qayta ishlash va kvant hisoblash kabi ko'plab sohalarni qamrab oladi. Kelajakdagi tadqiqotlar natijasida bitli algoritmlar nafaqat mavjud tizimlarning tezkorligini oshirish, balki yangi avlod texnologiyalarini yaratishda ham asosiy rol o'ynashi kutilmoqda. Shu bois, bitli algoritmlarni yanada chuqur o'rganish va ularni turli platformalarda kengroq qo'llash zamonaviy texnologik taraqqiyotning muhim qismidir.

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.Эрикссон. 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. C#. 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 C, C++, Java и Python, 2-е изд.: Пер. с англ. — СПб.: ООО "Альфа-книга", 2017. — 432 с.
9. Farmonov, S., & Nazirov, A. (2023). C# DASTURLASH TILIDA GRAY KODI BILAN ISHLASH. В CENTRAL ASIAN JOURNAL OF EDUCATION AND INNOVATION (Т. 2, Выпуск 12, сс. 71–74). Zenodo.
10. Farmonov, S., & Toirov, S. (2023). NETDA DASTURLASHNING ZAMONAVIY TEXNOLOGIYALARINI O'RGANISH. *Theoretical aspects in the formation of pedagogical sciences*, 2(22), 90-96
11. [Raxmonjonovich, F. S. \(2023\).](#) Array ma'lumotlar tizimini talabalarga o'qitishda Blockchain metodidan foydalanish. *Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari*, 2(2), 541-547.
12. [Raxmonjonovich, F. S. \(2023\).](#) Dasturlashda interfeyslardan foydalanishning ahamiyati. *Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari*, 2(2), 425-429.
13. [Raxmonjonovich, F. S. \(2023\).](#) Dasturlashda obyektga yo'naltirilgan dasturlashning ahamiyati. *Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari*, 2(2), 434-438.
14. [Raxmonjonovich, F. S. \(2023\).](#) Dasturlash tillarida fayllar bilan ishlash mavzusini Blended Learning metodi yordamida o'qitish. *Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari*, 2(2), 464-469.
15. [Raxmonjonovich, F. S. \(2023\).](#) DASTURLASHDA ISTISNOLARNING AHAMIYATI. *Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari*, 2(2), 475-481.
16. [Raxmonjonovich, F. S. \(2023\).](#) Dasturlashda abstraksiyaning o'rni. *Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari*, 2(2), 482-486.
17. [Raxmonjonovich, F. S., & Ravshanbek o'g'li, A. A. \(2023\).](#) Zamonaviy dasturlash tillarining qiyosiy tahlili. *Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari*, 2(2), 430-433.
18. [Raxmonjonovich, F. S. \(2023\).](#) C# dasturlash tilida fayl operatsiyalari qo'llashning qulayliklari haqida. *Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari*, 2(2), 439-446.

19. Raxmonjonovich, F. S. (2023). C# tilida ArrayList bilan ishlashning afzalliklari. *Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari*, 2(2), 470-474.

20. Farmonov Sherzodbek Raxmonjonovich, & Rustamova Humoraxon Sultonbek qizi. (2024). C# DASTURLASH TILIDA TO'PLAMLAR BILAN ISHLASH. Ta'lim Innovatsiyasi Va Integratsiyasi, 11(10), 210–214. Retrieved from <http://web-journal.ru/index.php/ilmiy/article/view/2480>.

21. Raxmonjonovich, F. S., & Ravshanbek o'g'li, A. A. (2023). Zamonaviy dasturlash tillarining qiyosiy tahlili. *Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari*, 2(2), 430-433.

22. Farmonov, S., & Rasuljonova, Z. (2024). OB'EKTGA YO'NALTIRILGAN DASTURLASH ZAMONAVIY DASTURLASHNING ASOSI SIFATIDA. *Центральноазиатский журнал образования и инноваций*, 3(1), 83-86.