MATEMATIKA VA HISOBLASHGA OID MASALALARNI BITLI ALGORITMLARI YORDAMIDA YECHISH

Farmonov Sherzodbek Raxmonovich
Farg‘ona davlat universiteti amaliy matematika va informatika kafedrasi katta o‘qituvchisi
farmonovsh@gmail.com
Abduqodirova Kamola Dilshodbek qizi
Farg‘ona davlat universiteti amaliy matematika yo‘nalishi 2-kurs talabasi
abduqodirovakamola463@gmail.com

Annotatsiya: Ushbu maqola bitli algoritmlar va ularning qo’llanilishi haqida yozilgan. Bitli algoritmlar, asosan, raqamli ma’lumotlarni kodlash va qayta ishlash jarayonida muhim rol o’ynaydi. Maqolaning asosiy maqsadi bitli algoritmlarning nazariy asoslarini, ularning turli xil turlari va amaliy qo’llanilishini tushuntirishdir. Asosiy bo’limlarda bitli operatsiyalar, masalan, AND (), OR (|), XOR (^), NOT (~) va bitli siljitish operatorlari (<<, >>) batafsil tahlil etiladi. C# da bitli ma’lumot turlari (masalan, byte, short, int, long) va ulardan foydalanish imkoniyatlari ham ko’rib chiqiladi.

Kalit so‘zlar: Bit manipulyatsiyasi, Bitlarni birlashtirish, Bitlarni aylantirish, Bit darajasida operatsiyalar, Samaradorlik, Xotira sarfi, Bitlar bilan ishlash, Bitni o’qish va yozish, Bitli masalalar, Hash funksiyalari, Bitli kodlash, Bitli algoritmlar samaradorligi, Dasturiy optimallashtirish.

Abstract: This article is written about bitwise algorithms and their applications. Bit algorithms mainly play an important role in the process of encoding and processing digital data. The main goal of the article is to explain the theoretical foundations of bit algorithms, their various types and practical applications. In the main sections, bitwise operations such as AND (), OR (|), XOR (^), NOT (~) and bitwise shift operators (<<, >>) are analyzed in detail. Bitwise data types (eg, byte, short, int, long) in C# and their uses are also covered.

Keywords: Bit manipulation, Bit concatenation, Bit rotation, Bitwise operations, Efficiency, Memory consumption, Working with bits, Bit problems, Hash functions, Bit encoding, Efficiency of bit algorithms, Software optimization.

Аннотация: Эта статья написана о побитовых алгоритмах и их приложениях. Битовые алгоритмы в основном играют важную роль в процессе кодирования и обработки цифровых данных. Основная цель статьи — объяснить теоретические основы битовых алгоритмов, их различные виды и практическое применение. В основных разделах подробно анализируются такие побитовые операции, как AND (), OR (|), XOR (^), NOT (~) и операторы побитового сдвига (<<, >>). Также рассматриваются побитовые типы данных (например, byte, short, int, long) в C# и их использование.

Ключевые слова: Битовые манипуляции, Конкатенация битов, Поворот битов, Побитовые операции, Производительность, Потребление памяти, Работа с битами, Битовые задачи, Хэш-функции, Кодирование битов, Эффективность битовых алгоритмов, Оптимизация программного обеспечения.

Bitli algoritmlar kompyuter fanida va dasturlashda juda samarali bo’lib, ular turli muammolarni hal qilishda ishlatiladi. Quyida ularning maqsadi, qo’llanilishi va qanday muammolarni hal qilishini batafsil tushuntiriladi.

Bitli algoritmlar nima? Bitli algoritmlar raqamlarni ikkilik sanoq sistemasida (0 va 1) ifodalash va shu formatda amallar bajarish usullarini anglatadi. Bu algoritmlar odatda juda samarali bo’lib, apparat darajasida tez ishlaydi.

Bitli algoritmlarining maqsadi. Bitli algoritmlardan foydalanishning asosiy maqsadlari quyidagilardan iborat:

Tezkor ishlash: Bitli operatsiyalar boshqa arifmetik yoki mantiqiy operatsiyalarga qaraganda ko’proq samaradorlikka ega.

Resurslarni tejash: Bitli operatsiyalar ko’pincha kamroq xotira va protsessor kuchi talab qiladi.

Past darajali manipulyatsiyalar: Uskunaga yaqin darajada ma’lumotlarni boshqarish uchun ishlatiladi.

Bitli algoritmi (Bitwise algorithm) raqamlarni bit darajasida ishlov berish chun qo'llaniladigan usullardir. Ular ko'plab sohalarda, jumladan:

1. Ma'lumotlar siqilishi: Bitli algoritmlar ma'lumotlarni siqish jarayonida samarali ishlatiladi. Masalan, Huffman kodlash va Run-Length Encoding (RLE) kabi usullar.
2. Kriptografiya: Bitli operatsiyalar shifrlash va ma'lumotlarni himoya qilishda muhim rol o'ynaydi. XOR operatsiyasi ko'pincha ma'lumotlarni shifrlashda qo'llaniladi.
3. Grafikalar va tasvirlar: Bitli algoritmlar piksel ma'lumotlarini manipulatsiya qilishda, masalan, rangni o'zgartirish yoki tasvirlarni filtr qilishda qo'llaniladi.

4. O'yin dasturlash: O'yinlarda harakatlar va ob'ektlar holatini saqlash uchun bitli baytlar ishlatilishi mumkin. Bu xotira samaradorligini oshiradi.
5. Tizim dasturlash: Operatsion tizimlar va past darajadagi dasturlar bitli operatsiyalarni ishlatadi, masalan, flaglar va rejimlarni boshqarishda.
6. Algoritmlar va ma'lumotlar tuzilmalari: Bitli algoritmlar ko'plab algoritmlarda, masalan, qidiruv va tartiblash algoritmlarida qo'llaniladi.
7. Matematika va hisoblash: Bitli operatsiyalar matematik hisob-kitoblarni tezlashtirishda, masalan, ko'paytirish yoki bo'lish jarayonlarida qo'llaniladi.

Umuman olganda, bitli algoritmlar kompyuter fanlari va dasturlashda juda muhim bo'lib, samaradorlikni oshirish va resurslardan unumli foydalanish imkonini beradi.
Matematika va hisoblash sohasida bitli algoritmlar turli masalalarda qo'llaniladi. Quyida ba'zi asosiy qo'llanilish sohalari keltirilgan:

1. Matematik operatsiyalar:

• Qo'shish va ayirish: Bitli algoritmlar qo'shish va ayirish operatsiyalarini tezlashtirish uchun ishlatiladi. Masalan, bitli qo'shish (XOR) yordamida raqamlarni qo'shish mumkin.

• Ko'paytirish va bo'lish: Bitli ko'paytirish algoritmlari, masalan, Booth algoritmi, raqamlarni tez ko'paytirish uchun ishlatiladi. Bo'lish jarayonida esa bitli bo'lish algoritmlari qo'llaniladi.

2. Qidiruv va tartiblash:

• Qidiruv algoritmlari: Bitli operatsiyalar yordamida qidiruv jarayonlari tezlashtirilishi mumkin. Masalan, bitli maskalar yordamida ma’lumotlar to’plamlarida qidiruvni amalga oshirish.

• Tartiblash algoritmlari: Bitli algoritmlar orqali ma’lumotlarni tartiblashda samaradorlik oshirilishi mumkin. Masalan, bitli tarmoq tartiblash algoritmlari.

3. Kriptografiya:

• Bitli operatsiyalar shifrlash va ma’lumotlarni himoya qilishda muhim rol o’ynaydi. XOR operatsiyasi ko’pincha simmetrik shifrlash algoritmlarida (masalan, AES) qo’llaniladi.

4. Hash funksiyalari:

• Hash funksiyalarida bitli operatsiyalar tezlik va samaradorlikni oshirish uchun ishlatiladi. Masalan, MD5 yoki SHA-1 kabi hash funksiyalarida bitli manipulyatsiyalar mavjud.

5. Matritsalar va vektorlar:

6. • Bitli algoritmlar matritsalardagi va vektorlardagi elementlarga tezkor operatsiyalarni amalga oshirishda qo’llaniladi. Masalan, bitli matritsa ko’paytirish.

7. Geometrik hisob-kitoblar:

• Geometrik masalalarda (masalan, nuqtalar orasidagi masofani hisoblash) bitli operatsiyalar yordamida tezkor hisob-kitoblar amalga oshirilishi mumkin.

8. Sonli tizimlar:

• Binom raqamlar tizimida operatsiyalarni amalga oshirishda bitli algoritmlar ishlatiladi. Masalan, decimal raqamlarni binar raqamlarga o’tkazishda.

Matematika va hisoblashda bitli algoritmlaridan foydalanish ko‘plab qiziqarli masalalarni yechishda qo‘llaniladi. Quyida matematika va hisoblashga oid masala keltirilgan:

Qo’shish va ayirish operatsiyalarini bitli algoritmlar yordamida tahlil qilish:

1. Qo’shish algoritmi

Qo’shish jarayoni quyidagi qadamlardan iborat:

1-qadam: XOR operatsiyasini qo’llash

• XOR (^) operatsiyasi yordamida har bir bitni qo'shadi. Bu operatsiya bitlar o'rtasidagi taqqoslashni amalga oshiradi:

• 0 ^ 0 = 0

• 0 ^ 1 = 1

• 1 ^ 0 = 1

• 1 ^ 1 = 0 (bu holda carry bo'ladi)

2-qadam: AND operatsiyasini qo'llash va chapga surish

• AND () operatsiyasi yordamida carry ni hisoblaydi. Bu carry ni chapga suradi (<<), chunki u keyingi bitga o'tadi.

3-qadam: Yangilash

• Yangilangan yig'indi va carry ni yangilab, jarayonni davom ettiradi. Agar carry nol bo'lsa, qo'shish to'xtaydi.

Qo'shish algoritmining kodda ko'rinishi

```
Static int BitwiseAdd(int a, int b)
{
  While (b != 0) // Carry nol bo'lmaguncha davom etadi
  {
    Int sum = a ^ b; // XOR operatsiyasi – yig'indini hisoblash
    B = (a & b) << 1; // AND operatsiyasi va chapga surish – carry ni hisoblaymiz
    A = sum; // Yig'indini yangilash
  }
  Return a; // Oxirgi natijani qaytarish
}
```

2. Ayirish algoritmi

Ayirish jarayoni quyidagi qadamlardan iborat:

1-qadam: Teskari qiymatini olish

• Ayirish uchun, birinchi raqamdan ikkinchi raqamni ayirish o'rniga, ikkinchi raqamning teskari qiymatini olish kerak. Buni ~ operatori (inversiya) va qo'shish yordamida amalga oshiradi:

• -b ni olish uchun b ni teskari qiymatga aylantiradi va unga 1 qo'shadi.

2-qadam: Qo'shish funksiyasini chaqirish

• Yuqorida keltirilgan qo'shish algoritmidan foydalanib, bunda birinchi raqamni ikkinchi raqamning teskari qiymati bilan qo'shadi.

Ayirish algoritmining kodda ko'rinishi

```
Static int BitwiseSubtract(int a, int b)
{
  B = BitwiseAdd(~b, 1); // Ikkinchi raqamning teskari qiymatini olish
  Return BitwiseAdd(a, b); // Birinchi raqamga teskari qiymatini qo'shish
}
```

Umumiy jarayon

Quyidagi kodda qo'shish va ayirishni birlashtirib ko'rsatiladi

```
Using System;
Class Program
{
  Static int BitwiseAdd(int a, int b)
  {
    While (b != 0)
    {
      Int sum = a ^ b; // XOR operatsiyasi
      B = (a & b) << 1; // AND va chapga surish
      A = sum; // Yig'indini yangilash
    }
    Return a; // Natijani qaytarish
  }
  Static int BitwiseSubtract(int a, int b)
  {
    B = BitwiseAdd(~b, 1); // Ikkinchi raqamning teskari qiymatini olish
    Return BitwiseAdd(a, b); // Birinchi raqamga teskari qiymatini qo'shish
  }
  Static void Main(string[] args)
```

```
    {
        Int num1 = 27; // Birinchi raqam
        Int num2 = 15; // Ikkinchi raqam
        Int addResult = BitwiseAdd(num1, num2);
        Console.WriteLine($"Qo'shish natijasi: {addResult}");
        Int subResult = BitwiseSubtract(num1, num2);
        Console.WriteLine($"Ayirish natijasi: {subResult}");
    }
}
```

Yuqoridagi algoritmlar bitli operatsiyalar yordamida qo'shish va ayirishni qanday amalga oshirishni batafsil ko'rsatildi.

Ushbu dasturda ikkita sonni qo'shadi va ayriydi.

```
using System;
Class Program
{
    Static void Main()
    {
        // Foydalanuvchidan ikkita sonni olish
        Console.Write("Birinchi sonni kiriting: ");
        Double son1 = Convert.ToDouble(Console.ReadLine());
        Console.Write("Ikkinchi sonni kiriting: ");
        Double son2 = Convert.ToDouble(Console.ReadLine());
        // Qo'shish
        Double qo'shishNatija = son1 + son2;
        Console.WriteLine($"Qo'shish natijasi: {qo'shishNatija}");
        // Ayirish
        Double ayirishNatija = son1 – son2;
        Console.WriteLine($"Ayirish natijasi: {ayirishNatija}");
        // Dastur tugagach, foydalanuvchidan tugmani bosishini kutamiz
        Console.WriteLine();
```

```
        Console.ReadLine();
    }
}
```

Natija:

Birinchi sonni kiriting: 10

Ikkinchi sonni kiriting: 5

Qo’shish natijasi: 15

Ayirish natijasi: 5

1. Using System; - Bu qator standart kutubxonani dasturga qo’shadi, bu esa Console klassidan foydalanish imkonini beradi.
2. Class Program – Dasturimizning asosiy sinfini belgilaydi. Har bir C# dasturi sinfdan boshlanadi.
3. Static void Main() – Dasturimizning boshlanish nuqtasi. Dastur ishga tushganda, eng avval shu metod bajariladi.
4. Foydalanuvchidan ma’lumot olish:

• Console.Write(“Birinchi sonni kiriting: “); - Foydalanuvchiga birinchi sonni kiritishni so’raydi.

• double son1 = Convert.ToDouble(Console.ReadLine()); - Foydalanuvchi kiritgan qiymatni o’qiydi va double turiga o’zgartiradi.

5. Ikkinchi sonni olish:

• Yuqoridagi jarayon ikkinchi son uchun ham takrorlanadi.

6. Qo’shish va natijani chiqarish:

• double qo’shishNatija = son1 + son2; - Ikkita sonni qo’shib natijani hisoblaydi.

• Console.WriteLine($”Qo’shish natijasi: {qo’shishNatija}”); - Qo’shish natijasini ekranga chiqaradi.

7. Ayirish va natijani chiqarish:

• double ayirishNatija = son1 – son2; - Ikkita sonni ayirib natijani hisoblaydi.

• Console.WriteLine($”Ayirish natijasi: {ayirishNatija}”); - Ayirish natijasini ekranga chiqaradi.

❖ Xulosa.

Ushbu masala orqali C# dasturlash tilida foydalanuvchidan ikkita sonni qabul qilib, ularni qo’shish va ayirish jarayonini amalga oshirish ko’rsatildi. Console.ReadLine() metodi yordamida foydalanuvchidan sonlarni kiritish so’raladi va Convert.ToDouble() yordamida bu qiymatlar double turiga o’zgartiriladi. Kiritilgan sonlar ustida qo’shish (+) va ayirish (-) operatsiyalari bajariladi. Hisoblangan natijalar Console.WriteLine() metodi yordamida ekranga chiqariladi, bu esa foydalanuvchiga natijalarni ko’rish imkonini beradi.

Foydalanilgan adabiyotlar ro‘yxati:

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. 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. B 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.

23. Farmonov, S., & Ro'zimatov, J. (2024). DASTURLASH TILLARINI O'RGANISHDA ONLINE TA'LIM PLATFORMALARIDAN FOYDALANISH. Theoretical aspects in the formation of pedagogical sciences, 3(1), 5-10.

24. Farmonov, S. R., & qizi Xomidova, M. A. (2024). C# VA JAVA DASTURLASH TILLARIDA FAYLLAR BILAN ISHLASHNING TURLI USULLARINING SAMARADORLIGI HAQIDA. *Zamonaviy fan va ta'lim yangiliklari xalqaro ilmiy jurnal*, *1*(9), 45-51.