

KOMPYUTER TARMOQLARI SOHASIDA BITLI ALGORITMLAR

Farmonov Sherzodbek Raxmonjonovich

*Farg'ona davlat universiteti amaliy matematika va
informatika kafedrasida katta o'qituvchisi*

farmonovsh@gmail.com

Xurshidaxon Nabiyeva Iqboljon qizi

Farg'ona davlat universiteti 2-kurs talabasi

xurshidanabiyeva00@gmail.com

Annotatsiya: Kompyuter tarmoqlari sohasida bitli algoritmlar, asosan, ma'lumotlarni uzatishda va tarmoqni xavfsiz qilishda yordam beradi. Ular quyidagi vazifalarni bajaradi. Ma'lumotlarni siqish: Bitli algoritmlar ma'lumotlarni siqishda ishlatiladi. Bu esa tarmoq orqali ma'lumotlar tezroq va kamroq hajmda uzatilishini ta'minlaydi. Shifrlash : Bitli algoritmlar ma'lumotlarni shifrlashda, ya'ni maxfiylikni saqlashda ishlatiladi. Masalan, tarmoq orqali uzatilayotgan ma'lumotlar begona kishilar tomonidan o'qilmasligi uchun shifrlanadi. Xatoliklarni aniqlash va tuzatish: Tarmoq orqali yuborilayotgan ma'lumotlarda xatolik bo'lishi mumkin. Bitli algoritmlar xatoliklarni aniqlash va tuzatish uchun ishlatiladi, bu esa ma'lumotlarning to'g'ri kelishini ta'minlaydi. Ma'lumotlarni paketlarga ajratish: Tarmoqdagi ma'lumotlar odatda kichik paketlarga bo'linadi. Bitli algoritmlar ushbu paketlarni yaratish, ularni manzilga yuborish va kerak bo'lsa, qayta tiklashda yordam beradi.

Kalit so'zlar: Raqamli tizim, algoritmlar, raqamli imzolar, protsessor, grafika, rangli kodlash, sun'iy intellekt, bitli algoritmlar, kompyuter tarmoqlari, bitli manipulyatsiya, shifrlash, arifmetik operatsiyalar, grafik interfeyslar, interaktiv tizimlar, signalni filtirlash.

Annotation: *In the field of computer networks, bit-level algorithms primarily assist in data transmission and ensuring network security. Data Compression: Bit-level algorithms are used for compressing data, which ensures that data is transmitted faster and in a smaller size across the network. Encryption: Bit-level algorithms are used to encrypt data, maintaining its confidentiality. For example, data transmitted across a network is encrypted to prevent unauthorized access by third parties. Error Detection and Correction: There may be errors in the data being transmitted over the network. Bit-level algorithms are used to detect and correct these errors, ensuring that the data arrives correctly. Packetization: Network data is typically divided into small packets. Bit-level algorithms assist in creating these packets, sending them to their destination, and, if necessary, reassembling them.*

Keywords: *Digital systems, algorithms, digital signatures, processor, graphics, color coding, artificial intelligence, bit-level algorithms, computer networks, bit manipulation, encryption, arithmetic operations, graphical interfaces, interactive systems, signal filtering.*

Аннотация: *В области компьютерных сетей битовые алгоритмы в основном помогают в передаче данных и обеспечении безопасности сети. Они выполняют следующие функции. Сжатие данных: Битовые алгоритмы используются для сжатия данных, что позволяет передавать информацию быстрее и с меньшим объемом по сети. Шифрование: Битовые алгоритмы применяются для шифрования данных, обеспечивая их конфиденциальность. Например, данные, передаваемые по сети, шифруются, чтобы посторонние не могли их прочитать. Обнаружение и исправление ошибок: В данных, передаваемых по сети, могут возникнуть ошибки. Битовые алгоритмы используются для их обнаружения и исправления, что обеспечивает правильность передаваемой информации. Разбиение данных на пакеты: Данные в сети обычно разделяются на маленькие пакеты. Битовые алгоритмы помогают создавать эти пакеты, отправлять их по назначению и при необходимости восстанавливать их.*

Ключевые слова: Цифровые системы, алгоритмы, цифровые подписи, процессор, графика, цветное кодирование, искусственный интеллект,

битовые алгоритмы, компьютерные сети, битовая манипуляция, шифрование, арифметические операции, графические интерфейсы, интерактивные системы, фильтрация сигналов.

Bitli algoritmlar (ya'ni, "binary algorithms") — bu raqamli tizimda ishlov beradigan va asosan ikkilik sanoq tizimidan (0 va 1) foydalanadigan algoritmlardir. Ular asosan kompyuterlar va raqamli tizimlarda, ayniqsa ma'lumotlar bazalari, kriptografiya, kompyuter tarmoqlari va ma'lumotlarni siqish kabi sohalarda qo'llaniladi.

Bitli algoritmlarning asosiy vazifalari:

1.Bitwise (bitga nisbatan) amallar: Bu amallar orqali ikkilik raqamlar ustida turli operatsiyalar bajariladi, masalan, AND, OR, XOR va NOT. Bu amallar juda tez va samarali hisoblashlarni ta'minlaydi.

AND (bitwise AND): Har ikkala bitning ham 1 bo'lishi kerak.

OR (bitwise OR): Bitlardan kamida biri 1 bo'lishi kerak.

XOR (bitwise XOR): Bitlar faqat bir xil bo'lmasligi kerak, ya'ni 1 va 0 yoki 0 va 1 bo'lishi kerak.

NOT (bitwise NOT): Bitning qiymatini teskari qilish (1 ni 0 ga, 0 ni 1 ga o'zgartirish).

2. Siqish algoritmlari: Ma'lumotni siqish jarayonida, bitlar asosida ishlov beriladi. Masalan, RLE (Run-Length Encoding) yoki Huffman coding algoritmlari, bitlar ketma-ketligini siqib, ma'lumotlarni kamaytiradi.

3. Kriptografiya: Bitli algoritmlar shifrlash va ma'lumotlarni xavfsiz yuborish uchun ishlatiladi. Masalan, AES (Advanced Encryption Standard) yoki RSA kabi kriptografik tizimlar bitli amallarni amalga oshiradi.

4. Tarmoqlar va protokollar: Kompyuter tarmoqlarida bitli algoritmlar tarmoq paketlarini qayta ishlashda va tarmoq protokollarini amalga oshirishda

muhim rol o'ynaydi. TCP/IP, Ethernet kabi protokollar, bitlarni uzatishda bitli manipulyatsiyalarni o'z ichiga oladi.

Bitli algoritmlar kompyuter tizimlarining tezligi va samaradorligini oshiradi, chunki ular kompyuterning ichki ishlash mexanizmlariga to'g'ri keladi va minimal resurslar bilan yuqori samarali hisoblashlarni amalga oshirishga imkon beradi.

Bitli algoritmlar turli sohalarda keng qo'llaniladi.

Ularning asosiy afzalligi — ularning ishlash tezligi va samaradorligi, chunki ular to'g'ridan-to'g'ri ikkilik tizimda (0 va 1) ishlaydi. Quyidagi sohalarda bitli algoritmlar eng ko'p qo'llaniladi:

1. Kriptografiya va xavfsizlik: shifrlash algoritmlari: Bitli algoritmlar ma'lumotlarni xavfsiz saqlash va uzatishda ishlatiladi. Masalan, AES (Advanced Encryption Standard) va RSA algoritmlarida bitwise amallar (AND, OR, XOR, NOT) xavfsiz shifrlash va dekodlash jarayonlarini amalga oshiradi. Hash funktsiyalari: MD5, SHA-256 kabi hash algoritmlari ma'lumotlarni tahlil qilish va xavfsizligini ta'minlashda bitli manipulyatsiyalarni qo'llaydi. Raqamli imzolar: Xavfsizlikni ta'minlashda va autentifikatsiyada bitli algoritmlar ishlatiladi.

2. Ma'lumotlarni siqish: siqish algoritmlari: Bitli algoritmlar ma'lumotlarni siqish (masalan, ZIP yoki PNG fayllarida) uchun ishlatiladi. Huffman coding va Run-Length Encoding (RLE) kabi siqish metodlarida bitlar yordamida ma'lumotlar optimal ravishda siqiladi. Rasm va video siqish: Tasvir va video fayllarini siqish uchun, masalan, JPEG yoki MPEG formatlarida, bitli manipulyatsiyalar qo'llaniladi.

3. Kompyuter tarmoqlari: tarmoq protokollari: Bitli algoritmlar kompyuter tarmoqlarida ma'lumotlarni uzatish va qabul qilishda ishlatiladi. TCP/IP, Ethernet kabi protokollar bitwise amallarni (masalan, bitni siljitish, masxlar) ishlatadi. Xatoliklarni tekshirish va tuzatish: Ma'lumotlar uzatilayotganda xatoliklarni aniqlash va tuzatish uchun bitli algoritmlar ishlatiladi. Masalan, CRC (Cyclic Redundancy Check) yoki paritet bitlari yordamida xatoliklarni aniqlash.

4. Kompyuter arxitekturasi va protsessorlar: tezkor hisoblashlar: Protsessorlarda bitli operatsiyalar yordamida tezkor hisoblashlar amalga oshiriladi. Bitwise amallar (masalan, bitni o'zgartirish, shifrlash, arifmetik operatsiyalar) tezlikni oshiradi. Bitwise masking va shifting: Bu amallar kompyuter arxitekturasida (masalan, 32-bit yoki 64-bit tizimlarda) tez-tez ishlatiladi.

5. Grafika va tasvirni qayta ishlash: tasvirni tahlil qilish: Bitli algoritmlar tasvirlardagi piksellarni qayta ishlashda qo'llaniladi. Rasmni siqish, tahrirlash yoki modifikatsiya qilishda bitlar yordamida manipulyatsiya qilinadi. Rangli kodlash: Tasvirlar va grafikalarini ko'rsatishda bitli algoritmlar ishlatiladi, masalan, tasvirni RGB formatida kodlash.

6. Mashina o'qitish va sun'iy intellekt: ma'lumotlarni tahlil qilish: Katta hajmdagi ma'lumotlarni tahlil qilishda bitli algoritmlar ishlatiladi. Sun'iy intellekt va mashina o'qitish tizimlari, masalan, neyron tarmoqlar, ma'lumotlarni kodlash va bitwise amallarni qo'llaydi. Xatoliklarni kamaytirish: Neyron tarmoqlar va o'qitish jarayonlarida bitli manipulyatsiyalar modelning samaradorligini oshiradi.

7. O'yin dasturlari va simulyatsiya: o'yin mexanikalari: O'yinlarda ob'ektlarni harakatlantirish, o'yin dunyosini yaratish va o'yin logikasini bajarish uchun bitli algoritmlar qo'llaniladi. Misol uchun, bitmasking usulida o'yin xaritalarini yaratish. Tasvirlarni qayta ishlash: 3D o'yinlarda va grafik simulyatsiyalarda tasvirni qayta ishlash va bitli algoritmlar yordamida optimallashtirish ishlari amalga oshiriladi.

8. Foydalanuvchi interfeyslari: grafik interfeyslar: Bitli algoritmlar foydalanuvchi interfeyslarini yaratishda ishlatiladi. Ekranda ko'rsatiladigan elementlar (masalan, tugmalar, ikonalar) bitlar yordamida tasvirlanadi. Interaktiv tizimlar: Interaktiv tizimlar va tizimlarning ishlashini optimallashtirishda bitli manipulyatsiyalar foydalaniladi.

9. Vaqtni sinxronlashtirish va signalni qayta ishlash: signalni filtrlash: Signalni qayta ishlashda bitli algoritmlar yordamida ma'lumotlarni filtrlaydi yoki o'zgartiradi. Masalan, audio yoki video signallarini yaxshilash uchun ishlatiladi.

Vaqtini sinxronlashtirish: Ovozli va vizual signallarni bir-biriga moslashtirishda ham bitli algoritmlar qo'llaniladi.

Bitli algoritmlar (binary algorithms) odatda ma'lumotlarni bit darajasida ishlashga asoslangan. Bunga misol sifatida, Bitwise AND (va), Bitwise OR (yoki), Bitwise XOR (xor), Bitwise NOT (inkor) kabi amallarni ko'rish mumkin.

C# dasturida ba'zi bitli amallarni amalga oshiradigan misolni keltiraman:

Misol: Bitli AND, OR, XOR va NOT amallarini amalga oshirish

using System;

class BitwiseOperations

{ static void Main() {

// 2 ta butun sonni olish

int a = 5; // 0101 (ikilik)

int b = 3; // 0011 (ikilik)

Console.WriteLine(\$"a = {a}, b = {b}");

// Bitwise AND (&)

int andResult = a & b;

Console.WriteLine(\$"Bitwise AND: {andResult}");

({Convert.ToString(andResult, 2)}");

// Bitwise OR (|)

int orResult = a | b;

Console.WriteLine(\$"Bitwise OR: {orResult}");

({Convert.ToString(orResult, 2)}");

// Bitwise XOR (^)

int xorResult = a ^ b;

Console.WriteLine(\$"Bitwise XOR: {xorResult}");

({Convert.ToString(xorResult, 2)}");

// Bitwise NOT (~)

int notResultA = ~a;

int notResultB = ~b;

```

Console.WriteLine($"Bitwise NOT of a: {notResultA}
({Convert.ToString(notResultA, 2)})");
Console.WriteLine($"Bitwise NOT of b: {notResultB}
({Convert.ToString(notResultB, 2)})");}

```

Masala tahlili:

1. Bitwise AND (&): Har bir bit bo'yicha faqat ikkala operandning bitlari 1 bo'lsa, natija 1 bo'ladi. Boshqa holatlarda 0 bo'ladi.

5 (0101) va 3 (0011) ning AND amali: $0101 \& 0011 = 0001$ (natija 1).

2. Bitwise OR (|): Agar ikkala operandning har bir bitida kamida bitta 1 bo'lsa, natija 1 bo'ladi.

5 (0101) va 3 (0011) ning OR amali: $0101 | 0011 = 0111$ (natija 7).

3. Bitwise XOR (^): Agar ikkala operandning har bir bitida bir xil bo'lmasa (ya'ni biri 1, boshqasi 0 bo'lsa), natija 1 bo'ladi.

5 (0101) va 3 (0011) ning XOR amali: $0101 \wedge 0011 = 0110$ (natija 6).

4. Bitwise NOT (~): Barcha bitlarni inkor qilish. Har bir 0 ni 1 ga, har bir 1 ni 0 ga aylantiradi.

5 (0101) ning NOT amali: $\sim 0101 = 1010$ (natija -6, chunki C#da ikkilik to'liq shakl 32-bit).

Chiqish:

a = 5, b = 3

Bitwise AND: 1 (1)

Bitwise OR: 7 (111)

Bitwise XOR: 6 (110)

Bitwise NOT of a: -6 (11111111111111111111111111111010)

Bitwise NOT of b: -4 (11111111111111111111111111111100)

Bu dastur bitli amallarni tushuntiruvchi oddiy misol bo'lib, bitli manipulyatsiyalarni

C#da qanday amalga oshirishni ko'rsatadi.

Xulosa: Kompyuter tarmog'larida bitli algoritmlar tarmoq samaradorligini oshirish, xavfsizlikni ta'minlash, xatoliklarni aniqlash va to'g'irlash, shifrlash va

siqish jarayonlarini optimallashtirishda muhim rol o'ynaydi. Ushbu algoritmlar tarmoq infratuzilmasini ishonchli va tezkor qilishda, shuningdek, tarmoq resurslarini samarali boshqarishda qo'llaniladi. Bitli algoritmlar tarmoq protokollarini qo'llab-quvvatlash, shuningdek, global tarmoqda xavfsiz va ishonchli ma'lumot uzatish uchun zarur vosita hisoblanadi.

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. *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* (T. 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.
25. Raxmonjonovich, F. S. (2024). C# VA MASHINA TILI. *Ta'lim innovatsiyasi va integratsiyasi*, 12(1), 59-62.
26. Farmonov, S. (2023). C# DASTURLASH TILIDA GRAY KODI BILAN ISHLASH. *Центральноазиатский журнал образования и инноваций*, 2(12 Part 2), 71-74.
27. Farmonov, S., & Jo'rayeva, M. (2023, December). DASTURLASHDA POLIMORFIZMNING AHAMIYATI. In *Международная конференция академических наук* (Vol. 2, No. 13, pp. 5-8).
28. Farmonov, S., & Usmonaliyev, U. (2024). O'ZBEKISTON RESPUBLIKASI IT SOHASINING RIVOJLANISH ISTIQBOLLARI. *Бюллетень педагогов нового Узбекистана*, 2(1), 59-62.