

MA'LUMOTLARNI OPTIMAL KODLASH VA SIQISHDA XOFMAN ALGORITMIDAN FOYDALANISH

Sh.R.Farmonov

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

farmonovsh@gmail.com

N.A.Muxamadaliyeva

Farg'ona davlat universiteti 2-kurs talabasi

abduqahorovanilufar306@gmail.com

Anotatsiya. Ushbu maqolada Xofman kodlash tizimi (Huffman Coding) ning asosiy prinsiplari va amaliyotdagi qo'llanilishi batafsil yoritiladi. Xofman kodlash algoritmi, 1952-yilda David A. Huffman tomonidan taklif qilingan va ma'lumotlarni siqishning samarali usullaridan biri sifatida keng qo'llanilmoqda. Maqolada Xofman kodlashining ishlash prinsipi, ya'ni, ma'lumotlarni siqishda minimal uzunlikdagi kodlarni qanday yaratish va optimal kodlash daraxtini qurish jarayoni tushuntiriladi.

Kalit so'zlar. Xofman kodlash, huffman coding, ma'lumotlarni siqish, kodlash tizimi, optimal kodlash daraxti, kompressiya algoritmi, dasturlash, ma'lumotlar uzatish, fayl siqish, kodlashning afzalliklari, ma'lumotlar strukturalari, xofman algoritmi, kompyuter ilm-fani, multimedia kompressiyasi, algoritmlar, rivojlanish istiqbollari.

Abstract. This article describes in detail the basic principles and practical application of the Huffman Coding system. The Huffman coding algorithm was proposed by David A. Huffman in 1952 and is widely used as an efficient data compression method. The article explains the working principle of Hofmann coding, i.e., how to create codes of minimum length in data compression and the process of building an optimal coding tree.

Keywords: Huffman coding, huffman coding, data compression, coding system, optimal coding tree, compression algorithm, programming, data transfer, file compression, advantages of coding, data structures, hoffman algorithm, computer science science, multimedia compression, algorithms, development perspectives.

Аннотация: В данной статье подробно описаны основные принципы и практическое применение системы кодирования Хаффмана. Алгоритм кодирования Хаффмана был предложен Дэвидом А. Хаффманом в 1952 году и широко используется как эффективный метод сжатия данных. В статье объясняется принцип работы кодирования Хаффмана, то есть способы создания кодов минимальной длины при сжатии данных и процесс построения оптимального дерева кодирования.

Ключевые слова: кодирование Хаффмана, кодирование Хаффмана, сжатие данных, система кодирования, оптимальное дерево кодирования, алгоритм сжатия, программирование, передача данных, сжатие файлов, преимущества кодирования, структуры данных, алгоритм Хаффмана, информатика, сжатие мультимедиа, алгоритмы, перспективы развития.

Ma'lumotlarni samarali siqish va uzatish zamonaviy kompyuter tizimlarining muhim vazifalaridan biridir. Bu jarayonlar ma'lumotlarni saqlash va uzatish resurslarini tejashga, shuningdek, tarmoqlar orqali tezkor va ishonchli ma'lumotlar almashish imkoniyatini yaratishga yordam beradi. Bunday tizimlardan biri bo'lgan Xofman kodlash (Huffman coding) algoritmi, ma'lumotlarni siqishda eng samarali va keng qo'llaniladigan usullardan biridir. 1952-yilda David A. Huffman tomonidan taklif qilingan ushbu algoritmi, ma'lumotlar elementlariga eng qisqa kodlar tayinlash orqali ma'lumotlar hajmini kamaytirishga imkon beradi.

Xofman kodlash algoritmining natijalari va samaradorligi

Xofman kodlash algoritmi, ma'lumotlarni siqish va kodlashda samarali natijalar beradigan metod sifatida tanilgan. Uning samaradorligi quyidagi aspektlarda ko'rinadi:

Kompressiya samaradorligi: Xofman kodlash tizimi eng qisqa kod uzunliklarini yaratishga qaratilgan bo'lib, bu uning asosiy afzalliklaridan biridir. Kodlash jarayonida ma'lumotlar ketma-ketligi tahlil qilinadi va har bir elementga ehtimoliyligi yuqori bo'lgan kodlar tayinlanadi, natijada fayl hajmi sezilarli darajada kamayadi. Xofman kodlashining samaradorligi eng yuqori ehtimollik bilan takrorlanadigan belgilarni qisqa kodlar bilan ifodalashda yotadi. Bu, ayniqsa, matn fayllari, rasm yoki audio fayllarini siqishda juda foydali.

Samaradorlik (tezlik): Xofman algoritmi asosan ikki bosqichdan iborat: birinchi bosqichda belgilar ehtimolligiga asoslangan daraxt quriladi, ikkinchi bosqichda esa bu daraxt yordamida har bir belgi uchun kodlar yaratiladi.

Hajmni kamaytirish: Xofman kodlash algoritmi odatda ma'lumotlarning hajmini 30-60% gacha kamaytirish imkonini beradi. Bu, ayniqsa, ko'p takrorlanadigan belgilarni o'z ichiga olgan ma'lumotlar uchun juda samarali. Misol uchun, matnli fayllarda, ayniqsa, tez-tez uchraydigan harflar bo'lsa (masalan, "e", "t", "a"), bu belgilar uchun qisqa kodlar tayinlanadi, bu esa siqish darajasini sezilarli darajada oshiradi.

Xofman kodlashining natijalari: Xofman kodlash algoritmi matnli fayllarni siqishda yuqori samaradorlikka erishadi. Masalan, ingliz tilidagi matnlar odatda 40-60% darajada siqilishi mumkin.

Rasm fayllari: Xofman kodlash rasm fayllarini siqish uchun ham qo'llaniladi, ammo u rasmning rang palitrasiga, piksellar ketma-ketligiga qarab samarali bo'lishi mumkin.

Audio fayllarini siqish: Xofman kodlash, ba'zan, audio fayllarda ham ishlatiladi (masalan, MP3 formatida). Biroq, video yoki audio fayllarda uni boshqa siqish usullari bilan birgalikda ishlatish kerak bo'lishi mumkin.

Tadqiqotning qisqacha natijasi: Ushbu tadqiqotda Xofman kodlash tizimi va uning amaliyotdagi qo'llanilishi tahlil qilindi. Xofman kodlash algoritmi, ma'lumotlarni samarali siqish va uzatish uchun eng optimal metodlardan biri sifatida o'z o'rnini topgan. Tadqiqotda ko'rsatilganidek, algoritm ma'lumotlar elementlarining ehtimollik taqsimotiga asoslanib, har bir belgi uchun minimal uzunlikdagi kodlarni yaratadi. Bu kodlash usuli fayl hajmini sezilarli darajada kamaytirishga imkon beradi va ko'plab sohalarda, jumladan, fayl siqish, multimedia kompressiyasi va internetda ma'lumotlarni tez uzatishda muvaffaqiyatli qo'llaniladi.

Xofman kodlash algoritmini C# dasturlash tilida amalga oshirish uchun quyidagi masala tuzishimiz mumkin:

Masala: Foydalanuvchi tomonidan kiritilgan matnni Xofman kodlash algoritmi yordamida siqib ko'rsatish. Dastur, har bir belgi uchun kodni yaratib, uni chiqarishi kerak. Shuningdek, kodlangan natijani saqlash va dekodlashni ham amalga oshirish.

C# dasturi:

```
using System;
using System.Collections.Generic;
using System.Linq;
class HuffmanNode
{
    public char Character { get; set; }
    public int Frequency { get; set; }
    public HuffmanNode Left { get; set; }
    public HuffmanNode Right { get; set; }
    public HuffmanNode(char character, int frequency)
    {
        Character = character;
        Frequency = frequency;
        Left = Right = null;
    }
}
class HuffmanCoding
{
    public static Dictionary<char, int> BuildFrequencyTable(string input)
    {
        Dictionary<char, int> frequencyTable = new Dictionary<char, int>();
        foreach (char character in input)
        {
            if (frequencyTable.ContainsKey(character))
                frequencyTable[character]++;
            else
                frequencyTable[character] = 1;
        }
        return frequencyTable;
    }
}
```

```
    }

    public static HuffmanNode BuildHuffmanTree(Dictionary<char, int>
frequencyTable)
    {
        PriorityQueue<HuffmanNode, int> priorityQueue = new
PriorityQueue<HuffmanNode, int>();
        foreach (var item in frequencyTable)
        {
            priorityQueue.Enqueue(new HuffmanNode(item.Key, item.Value),
item.Value);
        }
        while (priorityQueue.Count > 1)
        {
            HuffmanNode left = priorityQueue.Dequeue();
            HuffmanNode right = priorityQueue.Dequeue();
            HuffmanNode newNode = new HuffmanNode('\0', left.Frequency +
right.Frequency)
            {
                Left = left,
                Right = right
            };
            priorityQueue.Enqueue(newNode, newNode.Frequency);
        }
        return priorityQueue.Dequeue();
    }

    public static void GenerateCodes(HuffmanNode root, string currentCode,
Dictionary<char, string> codes)
    {
        if (root == null)
            return;
    }
```

```
        if (root.Character != '\0')
        {
            codes[root.Character] = currentCode;
        }
        GenerateCodes(root.Left, currentCode + "0", codes);
        GenerateCodes(root.Right, currentCode + "1", codes);
    }

    public static string Encode(string input, Dictionary<char, string> codes)
    {
        string encodedString = string.Empty;
        foreach (char character in input)
        {
            encodedString += codes[character];
        }
        return encodedString;
    }

    public static string Decode(string encodedString, HuffmanNode root)
    {
        HuffmanNode currentNode = root;
        string decodedString = string.Empty;
        foreach (char bit in encodedString)
        {
            currentNode = (bit == '0') ? currentNode.Left : currentNode.Right;
            if (currentNode.Left == null && currentNode.Right == null)
            {
                decodedString += currentNode.Character;
                currentNode = root;
            }
        }
        return decodedString;
    }
```

```
    }  
}  
  
class Program  
{  
    static void Main()  
    {  
        Console.WriteLine("Enter the text to be encoded:");  
        string input = Console.ReadLine();  
        Dictionary<char, int> frequencyTable =  
HuffmanCoding.BuildFrequencyTable(input);  
  
        HuffmanNode root =  
HuffmanCoding.BuildHuffmanTree(frequencyTable);  
        Dictionary<char, string> huffmanCodes = new Dictionary<char,  
string>();  
        HuffmanCoding.GenerateCodes(root, "", huffmanCodes);  
        Console.WriteLine("\nHuffman Codes for each character:");  
        foreach (var code in huffmanCodes)  
        {  
            Console.WriteLine($"Character: {code.Key} => Code:  
{code.Value}");  
        }  
        string encodedString = HuffmanCoding.Encode(input, huffmanCodes);  
        Console.WriteLine($"Encoded String: {encodedString}");  
        string decodedString = HuffmanCoding.Decode(encodedString,  
root);  
        Console.WriteLine($"Decoded String: {decodedString}");  
    }  
}
```

Natijalar:

Dastur quyidagi natijalarni ko'rsatadi:

Har bir belgi uchun Xofman kodi.

Kodlangan matn.

Dekodlangan (orijinal) matn.

Xofman kodlash tizimi ma'lumotlarni samarali siqish va uzatish uchun ishlatiladigan kuchli va optimal kodlash algoritmidir. Uning asosiy prinsiplaridan biri, tez-tez uchraydigan simvollarga qisqa kodlar, kam uchraydigan simvollarga esa uzunroq kodlar tayinlashdir. Bu orqali ma'lumotlarni maksimal darajada siqib, saqlash va uzatishning samaradorligini oshiradi.

Xofman kodlash tizimi amaliyotda keng qo'llaniladi, xususan, ma'lumotlarni siqish, telekommunikatsiya, ovozli va tasvirli ma'lumotlar kodlashda. Bu tizim ZIP arxivlash, JPEG rasm siqish, MP3 audio siqish kabi ko'plab texnologiyalarda muhim rol o'ynaydi. Shuningdek, u tarmoq uzatish samaradorligini oshirish va ma'lumotlarni kam joyda saqlashda qo'llaniladi.

Foydalanilgan adabiyotlar:

1. Huffman, D.A. (1952). "A Method for the Construction of Minimum-Redundancy Codes." *Proceedings of the IRE*, vol. 40, no. 9, pp. 1098–1101.
2. Knuth, D.E. (1997). *The Art of Computer Programming, Volume 3: Sorting and Searching*. Addison-Wesley.
3. Sayood, K. (2017). *Introduction to Data Compression*. Morgan Kaufmann.
4. Salomon, D. (2007). *Data Compression: The Complete Reference*. Springer.
5. Witten, I.H., Neal, R., & Cleary, J.G. (1987). Arithmetic Coding for Data Compression. *Communications of the ACM*.
6. Ziv, J., & Lempel, A. (1977). "A Universal Algorithm for Sequential Data Compression." *IEEE Transactions on Information Theory*, vol. 23, no. 3, pp. 337–343.
7. Moffat, A., & Zobel, J. (2006). Self-indexing Inverted Files for Fast Text Retrieval. *ACM Transactions on Information Systems*, 24(3), 289–319.
8. Tannenbaum, A.S. (2001). *Computer Networks* (4th ed.). Prentice Hall.

9. Harris, S., & Harris, D. (2012). Digital Design and Computer Architecture (2nd ed.). Morgan Kaufmann.
10. Raxmonjonovich, F. S. (2024). MINIMAL BOG‘LANISH DARAXTINI TOPISHDA PRIM ALGORITMIDAN FOYDALANISH. YANGI O‘ZBEKISTON, YANGI TADQIQOTLAR JURNALI, 1(3), 436-443.
11. Raxmonjonovich, F. S., & Kudratullo o‘g‘, K. U. B. (2024). C# VA NET FRAMEWORK ORQALI ZAMONAVIY VA XAVFSIZ TARMOQ DASTURLARINI ISHLAB CHIQISH. International journal of scientific researchers (IJSR) INDEXING, 5(2), 351-356.
12. Raxmonjonovich, F. S., & Azizjon o‘g‘li, N. A. (2024). WORKING WITH DATE AND TIME IN MODERN PROGRAMMING LANGUAGES. International journal of scientific researchers (IJSR) INDEXING, 5(2), 296-300.
13. Farmonov, S., & Rustamova, N. (2024, May). SINFLASHNING METRIK ALGORITMLARI, YAQIN QO‘SHNI USULI VA UNI UMUMLASHTIRISH HAMDA ULARNI NEYRON TARMOQ TEXNOLOGIYALARIDA QO‘LLANILISHI. In Международная конференция академических наук (Vol. 3, No. 5, pp. 71-75).
14. Farmonov, S., & Ergashaliyeva, B. (2024). QATIYMAS NEYRON TO‘RLAR: MAMDANI QATIYMAS MANTIQUIY XULOSASI, SUGENO QATIYMAS MANTIQUIY XULOSASI. Development and innovations in science, 3(5), 62-70.
15. Raxmonjonovich, F. S. (2024). XOFMAN KODLASH TIZIMI: AVIATSIYA VA PARVOZ MA'LUMOTLARINI SIQISHNING INNOVATSION YONDASHUVI. Лучшие интеллектуальные исследования, 34(1), 153-160.