

**MATINNI TAHLIL QILISH VA QIDIRISHDA BOYER MOORE  
ALGORITMINI QO'LANILISHI**

**Farmonov Sherzodbek Raxmonovich**

*Farmonov Sherzodbek Raxmonovich farg'ona davlat universiteti  
Amaliy matematika va informatika kafedrasida katta o'qituvchisi  
[farmonov@gmail.com](mailto:farmonov@gmail.com)*

**Pirimqulova Maftunaxon Axmadqul qizi**

*Fargona davlat universiteti 2-kurs talabasi  
[Jahongirpirimqulov01@gmail.com](mailto:Jahongirpirimqulov01@gmail.com)*

**Annotatsiya.** Ushbu maqolada Boyer-Moore algoritmining matnda naqsh (pattern)ni qidirish uchun juda samarali hisoblanadi. U matnni tahlil qilish va qidirishda quyidagi vazifalarda qo'llaniladi. Masalan: hujatlarni qidirish, qidiruv tizimlari, (search engines), kodni tahlil qilish (static code Analysis), log fayllarni taxlil qilish, plagiatni aniqlash, malumotlarni izlash tizimlari, real vaqt tizimida tahli qilish.

**Kalit so'zlar:** Boyer Moore, bing, web sahifa, Todo, Fixme, log fayllar, error, warning, palagiat, SQL injection naqishlar

**Annotation.** This article focuses on the Boyer-Moore algorithm, which is considered highly efficient for text pattern searching. It is used for text analysis and search in tasks such as document searching, search engines, code analysis (static code analysis), log file analysis, plagiarism detection, information retrieval systems, and real-time system analysis.

**Keywords:** Boyer-Moore, big, web page, Todo, Fixme, log files, error, warning, plagiarism, SQL injection patterns.

**Аннотация.** Данная статья посвящена алгоритму Бойера-Мура, который является очень эффективным для поиска текстовых шаблонов. Он применяется для анализа и поиска текста в следующих задачах: поиск в документах, поисковые системы, анализ кода (статический анализ кода), анализ лог-файлов, распознавание плагиата, системы поиска информации, анализ в режиме реального времени.

**Ключевые слова:** Бойер-Мур, big, веб-страница, Todo, Fixme, лог-файлы, ошибка, предупреждение, плагиат, шаблоны SQL-инъекций.

**Boyer-Moore** algoritmi - bu matnda kichik bir andozani (masalan, so'zni) qidirish uchun ishlatiladigan samarali qidiruv algoritmidir. U 1977-yilda Robert S. Boyer va J. Strother Moore tomonidan ishlab chiqilgan. Algoritm matnni o'ngdan chapga qarab

tekshirish orqali ishlaydi va ba'zi hollarda qidiruv jarayonini tezlashtiruvchi qoidalarni qo'llaydi.

Ключевые слова: Бойер-Мур, big, веб-страница, Todo, Fixme, лог-файлы, ошибка, предупреждение, плагиат, шаблоны SQL-инъекций.

### **Boyer Moore alqatitmi quyidagi sohalarda qolaniladi**

#### **1. Matnni qidirish tizimlari**

Matn muharrirlari (masalan, Microsoft Word, VS Code).

Katta hajmdagi hujjatlarda yoki ma'lumotlarda ma'lum so'z yoki iboralarni tez qidirish.

#### **2. Genetik tahlil**

DNK yoki RNK ketma-ketliklarida ma'lum gen yoki ketma-ketliklarni topish.

#### **3. Kod tahlili**

Katta hajmdagi kodlar ichida ma'lum bir funktsiyani yoki o'zgarmani topish.

Viruslar yoki xatolarni aniqlash.

#### **4. Kompilyatorlar va dasturiy tahlil**

Sintaksis tahlilida yoki operatorlarni izlashda samarali qo'llaniladi.

#### **5. Internet qidiruv tizimlari**

Qidiruv so'rovlarining tez ishlov berilishida yordam beradi.

#### **6. Ma'lumotlar xavfsizligi**

Fayl ichida zararli kod yoki shubhali qatorlarni topish uchun ishlatiladi.

#### **7. Tilni qayta ishlash (NLP)**

Matnlarda ma'lum iboralar yoki andozalarni qidirishda.

Nima uchun Boyer-Moore algoritmi samarali?

Boyer-Moore algoritmi matnni chapdan o'ngga emas, balki orqadan oldinga tekshiradi va sakrash mexanizmlaridan foydalanadi. Shu sababli, har bir qadamda ko'p belgilarni o'tkazib yuborishi va samaradorlikni oshirishi mumkin. Bu uni katta hajmdagi ma'lumotlarda ishlashda juda qulay qiladi.

#### **Boyer Moore afzaliklari quyidagicha:**

1. Samaradorlik: Uzun matnlarda qidiruvni optimallashtirish orqali tezlikni oshiradi.
2. Intellektual qidiruv: Matnni qidirishda oldindan tuzilgan qoidalar (masalan, naqshdagi oxirgi harfga asoslangan keyingi qadam) yordamida ishlaydi.
3. Moslashuvchanlik: Katta hajmdagi matnlar bilan samarali ishlaydi, chunki algoritmi keraksiz solishtirishlardan qochadi.

#### **Kamchiliklari**

1. Kichik shablon va matnlar uchun odiy algoritmlar kabi samarali emas
2. Oldindan tayorlash bosqichi nisbatan murakab

#### **Boyer moore algoritmi quyidagicha:**

##### **1. Kirish**

Matn (text): Qidiruv amalga oshiriladigan matn.

Andoza (pattern): Qidirilayotgan kichik matn.

## 2. Chiqarish

Matnda andoza topilgan joylarning indeksleri.

Algoritm bosqichlari:

## 1. Yomon belgi jadvalini tayyorlash

Matndagi har bir belgining oxirgi paydo bo'lgan joyini saqlaydigan jadvalni tuzish.

Agar belgi jadvalda topilmasa, unga standart qiymat (-1) beriladi.

## 2. Moslikni tekshirish

Andoza o'ngdan chapga qarab matn bilan solishtiriladi.

Agar biron belgi mos kelmasa, andoza yomon belgi jadvaliga ko'ra siljtiladi.

## 3. Siljitish qoidasi

Agar moslik topilsa:

Indeks saqlanadi va andoza davomida siljiydi.

Agar moslik topilmasa:

Andoza yomon belgi jadvaliga ko'ra kerakli miqdorda o'ngga siljtiladi.

**Boyer Moore algoritmi yordamida malum soz yoki iborani qidiruvchi masala**

**Masala:** Matinda "Algoritm" so'zni qidiring. Agar so'z topilmasa, tegishli habarni chiqaring

```
using System;
```

```
using System.Collections.Generic;
```

```
class BoyerMooreSearch
```

```
{
```

```
// Noto'g'ri xarakter jadvalini tayyorlash
```

```
static Dictionary<char, int> PreprocessBadCharacter(string pattern)
```

```
{
```

```
    var badCharTable = new Dictionary<char, int>();
```

```
    int patternLength = pattern.Length;
```

```
    for (int i = 0; i < patternLength; i++)
```

```
    {
```

```
        badCharTable[pattern[i]] = i;
```

```
    }
```

```
    return badCharTable;
```

```
}
```

```
// Boyer-Moore algoritmi
```

```
static List<int> Search(string text, string pattern)
```

```
{
```

```
    var badCharTable = PreprocessBadCharacter(pattern);
```

```
    int textLength = text.Length;
```

```
    int patternLength = pattern.Length;
```

```
    var indices = new List<int>();
```

```
    int shift = 0;
```

```

while (shift <= textLength - patternLength)
{
    int j = patternLength - 1;
    while (j >= 0 && pattern[j] == text[shift + j])
    {
        j--;
    }
    if (j < 0)
    {
        indices.Add(shift);
        shift += (shift + patternLength < textLength) ? patternLength -
badCharTable.GetValueOrDefault(text[shift + patternLength], -1) : 1;
    }
    else
    {
        shift += Math.Max(1, j - badCharTable.GetValueOrDefault(text[shift + j], -
1));
    }
}
return indices;
}
// Rang bilan ajratish
static void HighlightText(string text, List<int> indices, string pattern)
{
    int patternLength = pattern.Length;
    int lastIndex = 0;
    foreach (var index in indices)
    {
        Console.Write(text.Substring(lastIndex, index - lastIndex));
        Console.ForegroundColor = ConsoleColor.Red; // Rangni o'zgartirish
        Console.Write(text.Substring(index, patternLength));
        Console.ResetColor();
        lastIndex = index + patternLength;
    }
    Console.WriteLine(text.Substring(lastIndex)); // Qolgan matnni chiqarish
}

static void Main(string[] args)
{
    // Matn va qidiriladigan so'z

```

```
string text = "Boyer-Moore algoritmi samarali va tez ishlaydi. Bu algoritm katta matnlarda ishlatiladi.";
```

```
string pattern = "algoritm";
```

```
Console.WriteLine("Matn: " + text);
```

```
Console.WriteLine("\nQidirilayotgan so'z: " + pattern);
```

```
var indices = Search(text, pattern);
```

```
if (indices.Count > 0)
```

```
{ Console.WriteLine("\nTopilgan joylar:");
```

```
    HighlightText(text, indices, pattern);}
```

```
else
```

```
{ Console.WriteLine("\nSo'z yoki ibora topilmadi."); }
```

```
}
```

```
}
```

### Chiqish natijasi;

Boyer-Moore algoritmi samarali va tez ishlaydi. Bu algoritm katta matnlarda ishlatiladi.

Qidirilayotgan so'z: algoritm

**Umumiy masalalar uchun dastur** : Katta matndan bir nechta andozalarni qidirish

**Masala ta'rifi**: Bir matn berilgan va bizdan bu matndan bir nechta andozalarni topish talab qilinadi. Har bir andozaning matndagi barcha topilgan indekslarini chiqarish kerak.

**Masala yechimi**:

1. Katta matn (text) va bir nechta andozalar (patterns) berilgan bo'lib, har bir andozaning matn ichidagi joylarini topish kerak.

2. Boyer-Moore algoritmini har bir andoza uchun alohida ishlatamiz.

3. Har bir topilgan andozaning boshlanish joylarini natija sifatida chiqaramiz.

Endi bu masalani dasturini tuzamiz :

```
using System;
```

```
using System.Collections.Generic;
```

```
class BoyerMooreSearch
```

```
{// Yomon belgi jadvalini yaratish funksiyasi
```

```
    public static Dictionary<char, int> CreateBadCharTable(string pattern)
```

```
    { var badChar = new Dictionary<char, int>();
```

```
        int m = pattern.Length;
```

```
    // Jadvalni to'ldirish
```

```
        for (int i = 0; i < m; i++)
```

```
            { badChar[pattern[i]] = i; // Belgining eng oxirgi indeksini saqlaydi } return
```

```
badChar;
```

```
}
```

```
// Boyer-Moore algoritmi asosiy funksiyasi
```

```
    public static List<int> BoyerMoore(string text, string pattern)
```

```

{ List<int> result = new List<int>();
  int n = text.Length;
  int m = pattern.Length;
// Yomon belgi jadvali
  var badChar = CreateBadCharTable(pattern);
  int shift = 0; // Matn bo'yicha qidiruv
  while (shift <= (n - m))
  { int j = m - 1;
// Matn va andozani o'ngdan chapga qarab solishtirish
    while (j >= 0 && pattern[j] == text[shift + j])
      { j-- }
// Agar moslik topilsa
    if (j < 0)
      { result.Add(shift); // Indeksni saqlash
// Keyingi siljitish
        shift += (shift + m < n) ? m - badChar.GetValueOrDefault(text[shift + m], -1) :
1 }
    else
      { // Moslik yo'q: yomon belgi qoidasi bo'yicha siljitish
        shift += Math.Max(1, j - badChar.GetValueOrDefault(text[shift + j], -1)); } }
return result;
}
// Bir nechta andozalarni qidirish
public static void SearchPatterns(string text, List<string> patterns)
{ foreach (var pattern in patterns)
  { Console.WriteLine($"Qidirilayotgan andoza: {pattern}");
// Boyer-Moore algoritmi yordamida qidiruv
    List<int> positions = BoyerMoore(text.ToLower(), pattern.ToLower());
// Natijalarni chiqarish
    if (positions.Count > 0)
      { Console.WriteLine("Andoza quyidagi indekslarda topildi:");
        foreach (int pos in positions)
          { Console.WriteLine(pos)}
        else
          { Console.WriteLine("Andoza matnda topilmadi.");}
      }
} // Asosiy dastur
public static void Main(string[] args)
{ // Matn va andozalar
  string text = "Boyer-Moore algoritmi yordamida matndan bir nechta andozalarni

```

qidiramiz. Algoritmi samarali va tez.";

```
List<string> patterns = new List<string> { "algoritmi", "yordamida", "matndan",  
"tez" };
```

```
// Qidiruvni amalga oshirish
```

```
SearchPatterns(text, patterns);  
}  
}
```

**Dastur tahlili;** Ushbu dastur Boyer-Moore algoritmini quyidagi maqsad uchun ishlatadi: Bir nechta andozalarni qidirish: Matndan bir nechta andozalarni izlaydi va ularning boshlanish joylarini chiqaradi.

Samaradorlik: Algoritm andoza va matnni faqat kerakli holatlarda taqqoslaydi, bu esa qidiruvni sezilarli darajada tezlashtiradi.

Dastur ishlash jarayoni:

1. Matndan qidiriladigan andozalar ro'yxati olinadi.

2. Har bir andoza uchun Boyer-Moore algoritmi ishlaydi:

Yomon belgi jadvali: Algoritm matnni nechta qadamga siljitish kerakligini aniqlaydi.

Taqqoslash: Andoza bilan matnning mos keladigan qismlari topiladi.

Topilgan joylar saqlanadi.

3. Natijalar rangli ko'rinishda foydalanuvchiga taqdim etiladi:

Andoza bo'lagi matnda qizil rangda ajratiladi.

Topilgan indekslar yashil rangda ko'rsatiladi.

### 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. 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 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>.