
**MATNLARNI PLAGIATGA TEKSHIRISHDA BOYER-MOORE ALGORITMINING
TEZKOR VA SAMARALI QO'LLANILISHI**

Farmonov Sherzodbek RaxmonjonovichFarg'ona davlat universiteti amaliy matematika va
informatika kafedrası katta o'qituvchisifarmonov@gmail.com**Ma'murova Mashhuraxon Rahmatjon qizi**Farg'ona davlat universiteti 2-kurs talabasi.
mamurovmansurbek81@gmail.com

Annotatsiya:Maqola Boyer-Moore algoritmining plagiatni aniqlashda samaradorligini o'rganadi. Boyer-Moore algoritmi katta matnlar bilan ishlashda minimal vaqt sarf qiladi, shu bois plagiat aniqlashda foydalidir. Maqolada algoritmnining amaliyotda qo'llanilishi va uning boshqa qidiruv metodlari bilan taqqoslanishi haqida misollar keltirilgan.

Kalit so'zlar: Boyer-Moore algoritmi, matn qidirish, matnni plagiatga tekshirish, algoritmik samaradorlik, plagiatga qarshi tizimlar, matnlarni taqqoslash, plagiatni oldini olish, heuristika.

Annotation: This paper studies the effectiveness of the Boyer-Moore algorithm in plagiarism detection. The Boyer-Moore algorithm is efficient in processing large texts with minimal time consumption, making it useful for plagiarism detection. The paper provides examples of the algorithm's practical application and compares it with other search methods.

Keywords: Boyer-Moore algorithm, text search, plagiarism detection, algorithmic efficiency, anti-plagiarism systems, text comparison, plagiarism prevention, heuristics.

Аннотация: В статье исследуется эффективность алгоритма Бойера-Мура для обнаружения плагиата. Алгоритм Бойера-Мура эффективно работает с большими текстами, минимизируя время обработки, что делает его полезным для обнаружения плагиата. В статье приведены примеры практического применения алгоритма и его сравнение с другими методами поиска.

Ключевые слова: алгоритм Бойера-Мура, поиск по тексту, проверка текста на плагиат, алгоритмическая эффективность, системы защиты от плагиата, сравнение текстов, предотвращение плагиата, эвристика.

Boyer-Moore algoritmi - matn ichidan ma'lum bir pattern (so'z yoki fraza)ni qidirish uchun ishlatiladigan samarali algoritmdir. Bu algoritmi 1977-yilda Robert S. Boyer va J. Strother Moore tomonidan ishlab chiqilgan va matnni qidirish (pattern matching) sohasida eng tez ishlaydigan metodlardan biri hisoblanadi.

Algoritmning asosiy g'oyasi:

Boyer-Moore algoritmi matndagi patternni (so'z yoki fraza) topishda ishlatiladi va quyidagi yondashuvlardan foydalangan holda matnni tezda tahlil qilish imkonini beradi:

1. Matnni o'ngdan chapga qarab tekshirish: Boyer-Moore algoritmi, an'anaviy qidiruv usullaridan farqli o'laroq, patternni matn bo'ylab o'ngdan chapga qarab taqqoslaydi. Bu yondashuv ko'proq mos kelmasliklarni erta aniqlashga yordam beradi.

2. Heuristikalar yordamida optimallashtirish: Bad Character Heuristic (Yomon Belgilar Heuristikasi): Agar pattern va matnning mos kelmaydigan belgilari topilsa, patternni mos kelmaydigan belgidan keyin siljitish. Bu jarayon ko'plab tekshiruvlarni o'tkazmasdan natijaga erishishga yordam beradi.

Good Suffix Heuristic (Yaxshi Sufiks Heuristikasi): Agar patternning bir qismi matn bilan mos kelsa, qolgan qismini qayta tekshirishni kamaytirish uchun patternni shu qismni moslashtirib siljitish.

Qo'llanish sohalari:

1. Qidiruv tizimlari (Search Engines):
2. Plagiatni aniqlash (Plagiarism Detection)
3. Biologik tahlil (Bioinformatics)
4. Kompilyatorlar va tahlilchilar (Compilers and Parsers)
5. Indeksatsiya va ma'lumotlar bazasida qidiruv (Database Search and Indexing)
6. Kriptografiya va xavfsizlik (Cryptography and Security)
7. O'yinlar va simulyatsiyalar (Games and Simulations)

Plagiatni aniqlashda Boyer-Moore algoritmining qo'llanilishi - bu matnli hujjatlar orasidagi takrorlanishlarni aniqlash va plagiatni tekshirish uchun samarali usul hisoblanadi. Boyer-Moore algoritmi matnni qidirishda yuqori samaradorlikka ega bo'lgani sababli, u plagiatni aniqlash tizimlarida juda foydalidir, ayniqsa, katta hajmdagi matnlar bilan ishlashda.

Boyer-Moore algoritmining plagiatni aniqlashdagi roli.

Plagiatni aniqlash - bu original yoki o'zgartirilgan matnlar orasidagi o'xshashlikni topish jarayonidir. Boyer-Moore algoritmi bu jarayonda quyidagi tarzda ishlaydi:

1. Matnni qidirish.
2. Katta matnlar bilan ishlash.
3. Heuristikalar yordamida tezlikni oshirish.

Boyer-Moore algoritmi plagiatni aniqlashda qanday qo'llaniladi?

1. Matnni taqqoslash: Boyer-Moore algoritmi yordamida matnli hujjatlar orasida so'z va frazalarni taqqoslash osonlashadi. Bu, ayniqsa, internetdagi maqolalar, blog postlar, va ilmiy ishlarda ko'p takrorlanuvchi frazalarni aniqlashda foydalidir.
2. Shaxsiy va akademik tekshiruvlar: Boyer-Moore algoritmi universitetlarda yoki ilmiy tadqiqot markazlarida plagiatni tekshirishda ishlatiladi. Talabalar yoki tadqiqotchilar tomonidan yuborilgan matnlarning boshqa manbalar bilan o'xshashligi Boyer-Moore algoritmi orqali aniqlanadi.
3. Veb sahifalar va onlayn materiallarni tekshirish: Internetdagi yoki onlayn resurslardagi matnlar orasidagi o'xshashliklarni topishda Boyer-Moore algoritmi ishlatiladi. Bu plagiatni aniqlash tizimlarining asosiy vositalaridan biridir.

Masala : Aytaylik, sizda quyidagi ikkita matn bor:

Matn 1: Boyer-Moore algoritmi matnni qidirish sohasidagi eng samarali algoritmlardan biridir.

Matn 2: Boyer-Moore algoritmi matnni qidirishda tez va samarali ishlaydi.

Maqsad: Bu ikkita matn orasidagi o'xshashlikni Boyer-Moore algoritmi yordamida aniqlash va matndagi o'xshashlik qayerda boshlanayotganligini topish.

C# dasturidagi yechimi:

```
using System;
```

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

```
class BoyerMoorePlagiarism
```

```
{
```

```
    // Bad Character Heuristic funksiyasi
```

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

```
    {
```

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

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

```
        // Patternning har bir belgi uchun bad character heuristicni hisoblash
```

```
        for (int i = 0; i < m - 1; i++)
```

```

    {
        badChar[pattern[i]] = m - 1 - i;
    }
    return badChar;
}
// Boyer-Moore algoritmi yordamida matni plagiatsiz tekshirish
public static int BoyerMooreSearch(string text, string pattern)
{
    int n = text.Length;
    int m = pattern.Length;
    // Agar pattern matndan uzunroq bo'lsa, topish mumkin emas
    if (m > n)
    {
        return -1;
    }
    // Bad character heuristicni yaratish
    var badChar = BadCharacterHeuristic(pattern);
    int i = 0; // Matn bo'ylab qidirishni boshlash indeksi
    while (i <= n - m)
    {
        int j = m - 1;
        // Matn va patternni o'ngdan chapga tekshirish
        while (j >= 0 && pattern[j] == text[i + j])
        {
            j--;
        }
        // Agar pattern to'liq mos kelsa, boshlanish indeksini qaytarish
        if (j == -1)
        {
            return i; // Pattern topildi, boshlanish indeksi
        }
        else
        {
            // Bad character heuristic asosida i ni yangilash
            int shift = badChar.ContainsKey(text[i + j]) ? badChar[text[i + j]] : m;
            i += shift;
        }
    }
    // Agar pattern topilmasa
    return -1;
}
public static void Main(string[] args)
{
    // Asl matn (pattern) va tekshirilayotgan matn (text)
    string text = "Boyer-Moore algoritmi matni qidirishda tez va samarali ishlaydi.";
    string pattern = "Boyer-Moore algoritmi matni qidirish sohasidagi eng samarali
    algoritmlardan biridir.";
}

```

```

// Plagiatni tekshirish
int result = BoyerMooreSearch(text.ToLower(), pattern.ToLower());
// Natijani chiqarish
if (result != -1)
{
    Console.WriteLine($"Plagiat topildi. Pattern boshlanishi: {result}");
}
else
{
    Console.WriteLine("Plagiat topilmadi.");
}
}
}

```

Kodni Tushunish:

1. Bad Character Heuristic: Bu funksiya pattern (manba matn)ning har bir belgisini tekshiradi va ularni qidirish jarayonini optimallashtiradi. Agar patternning biron-bir belgisini topmasa, qidiruvni qanchalik bosqichni o'tkazib yuborish kerakligini hisoblaydi.

2. Boyer-Moore Search: BoyerMooreSearch funksiyasi matn va patternni o'ngdan chapga qarab tekshiradi. Pattern matnda to'liq mos kelsa, boshlanish indeksini qaytaradi. Agar patternning har bir belgisi matnda to'g'ri kelmasa, Bad Character Heuristic yordamida qidiruv jarayoni optimallashtiriladi.

3. Natija chiqarish: Agar pattern matnda topilsa, boshlanish indeksini chiqaradi (qayerdan boshlanishi), aks holda "Plagiat topilmadi" deb xabar chiqaradi.

Pattern (Matn 1): Boyer-Moore algoritmi matnni qidirish sohasidagi eng samarali algoritmlardan biridir.

Text (Matn 2): Boyer-Moore algoritmi matnni qidirishda tez va samarali ishlaydi.

Natija: Plagiat topilmadi.

Izoh: Boyer-Moore algoritmi matnlarni qidirishda samarali ishlaydi, chunki u faqat kerakli hollarda belgilarni tekshiradi. Ammo, ikkala matn o'rtasida ba'zi o'xshashliklar mavjud bo'lsa ham, ular to'liq mos kelmaydi, shuning uchun plagiat topilmadi.

Agar sizda boshqa manbadan olingan matn bilan solishtirish uchun plagiatni aniqlash tizimi kerak bo'lsa, pattern sifatida original matndan yoki shubhali manbadan olingan fraza (so'z)ni kiritishingiz mumkin. Bu kod yordamida siz matndagi o'xshashliklarni aniqlay olasiz.

Plagiatni aniqlashda Boyer-Moore algoritmining afzalliklari.

1. Tezlik va samaradorlik: Boyer-Moore algoritmi katta matnlar orasida so'z yoki frazalarni qidirishda juda tez ishlaydi. Bu plagiatni aniqlash tizimlarida juda foydalidir, chunki ular ko'pincha katta hajmdagi matnlarni tahlil qilishni talab qiladi.

2. O'xshashliklarni aniqlash: Boyer-Moore algoritmi faqat to'liq mos keladigan patternlarni topish bilan cheklanmaydi, balki plagiat tizimida to'liq yoki qisman mos keladigan bo'laklarni aniqlashga yordam beradi.

3. Keng qo'llanilish doirasi: Algoritm nafaqat matnli fayllarda, balki kodlashda ham ishlatiladi. Masalan, dasturlash tilida yozilgan matnlar orasidagi o'xshashliklarni topish uchun ham ishlatilishi mumkin.

Plagiatni aniqlashda Boyer-Moore algoritmining kamchiliklari.

1. Kichik patternlar uchun samaradorlikning pastligi: Agar pattern juda kichik bo'lsa, Boyer-Moore algoritmi boshqa algoritmlar (masalan, Knuth-Morris-Pratt yoki Rabin-Karp)ga

qaraganda kamroq samarali bo'lishi mumkin. Shuning uchun juda kichik so'zlar yoki frazalarda ishlashda ba'zi holatlarda samaradorlik pasayishi mumkin.

2. Qayta tekshiruvlar: Boyer-Moore algoritmi ba'zida faqat to'liq mos keladigan patternlarni qidiradi, lekin plagiatni aniqlashda ba'zan qisman mos keladigan o'xshashliklar ham aniqlanishi kerak. Shu sababli, ba'zi modifikatsiyalar talab qilinishi mumkin.

Boyer-Moore algoritmining afzalliklari uning samaradorligi va ishlash tezligida yotadi, ayniqsa katta hajmdagi matnlar bilan ishlaganda. Shuningdek, algoritmining heuristikalar asosidagi yondashuvi nafaqat vaqtni, balki resurslarni ham tejaydi, bu esa uni real va amaliy ilovalarda qo'llash uchun juda foydali qiladi.

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*. 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. Центральноеазиатский 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. Центральноеазиатский 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.
29. Raxmonjonovich, F. S., & Xasan o'g'li, X. O. (2023). DASTURLASHDA SANA VA VAQTLAR BILAN ISHLASH. Ta'lim innovatsiyasi va integratsiyasi, 11(11), 3-6.