



MATNNI SAMARALI QIDIRISHNING INSONIYATGA QO'SHGAN HISSASI

Sh.R.Farmonov

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

farmonovsh@gmail.com

H.A.Ismoilova

Farg'ona davlat universiteti 2-kurs talabasi

xulkaroyismoilova443@gmail.com

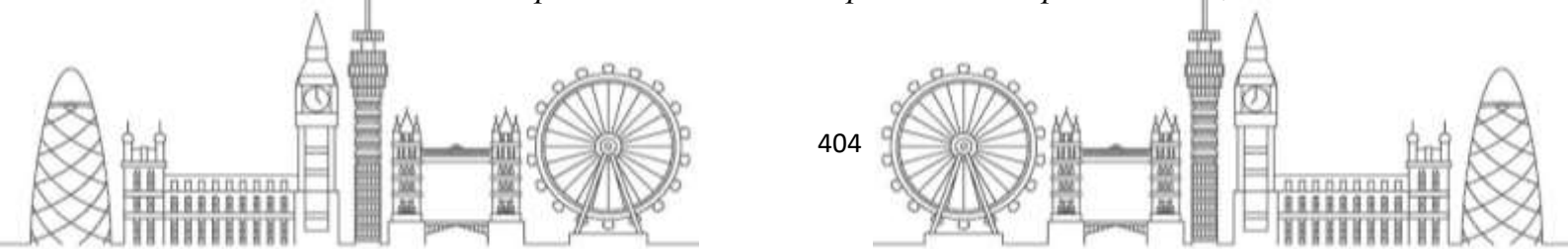
Annotatsiya: Knut-Morris-Prat (KMP) algoritmi, matn qidirish sohasida eng samarali va tezkor metodlardan biri hisoblanadi. Ushbu maqola KMP algoritmining ishlash printsipli, afzalliklari va qo'llanilish sohalarini ko'rib chiqadi. Algoritm matn ichidagi submatnni faqat bir marta tekshirib, takroriy ishlarni minimallashtirishga asoslangan. Bu maqolada KMP algoritmining asosiysi, uni real hayotdagi dasturlarda, bioinformatika, versiya boshqaruvi tizimlari va siqish algoritmlarida qanday ishlatilishi haqida so'z yuritiladi. KMP algoritmining samaradorligi va qo'llanilish sohasi uning umumiy kompyuter fanlari sohasida keng tarqalganligini ko'rsatadi.

Kalit so'zlar: KMP algoritmi, matn qidirish, samaradorlik, bioinformatika, versiya boshqaruvi, siqish algoritmlari, matnni tahlil qilish, submatnni topish, algoritmik samaradorlik, preprocessing, lps massivi, kompyuter fanlari.

Annotation: The Knut-Morris-Prat (KMP) algorithm is one of the most efficient and fast methods for string matching. This article discusses the working principle, advantages, and areas of application of the KMP algorithm. The algorithm is based on minimizing repetitive operations by checking the text only once. The article explores the use of the KMP algorithm in real-world applications such as bioinformatics, version control systems, and compression algorithms. The efficiency of the algorithm and its widespread use in the field of computer science demonstrate its significance.

Keywords: KMP algorithm, string matching, efficiency, bioinformatics, version control systems, compression algorithms, text analysis, substring search, algorithmic efficiency, preprocessing, lps array, computer science.

Аннотация: Алгоритм Knut-Morris-Prat (KMP) является одним из самых эффективных и быстрых методов для поиска строк. В данной статье рассматриваются принцип работы алгоритма KMP, его преимущества и области применения. Алгоритм основывается на том, что он проверяет текст только один раз, минимизируя повторяющиеся операции. В статье обсуждается использование алгоритма KMP в реальных приложениях, таких как





MODERN PROBLEMS IN EDUCATION AND THEIR SCIENTIFIC SOLUTIONS

биоинформатика, системы управления версиями и алгоритмы сжатия. Эффективность алгоритма и его широкое применение в области компьютерных наук подчеркивают его значимость.

Ключевые слова: Алгоритм KMP, поиск строк, эффективность, биоинформатика, системы управления версиями, алгоритмы сжатия, анализ текста, нахождение подстроки, алгоритмическая эффективность, preprocessing, массив lps, компьютерные науки.

Knut-Morris-Prat (KMP) algoritmi, matnni qidirish (string matching) muammosini samarali va tezkor tarzda hal qilish uchun mo'ljallangan klassik algoritmdir. 1977 yilda Donald Knut, Vaughan Pratt va James H. Morris tomonidan taqdim etilgan bu algoritm, matn ichida belgilangan submatnni topish jarayonini optimallashtirishda asosiy vosita sifatida qo'llaniladi. KMP algoritmi boshqa ko'plab matn qidirish algoritmlaridan farq qiladi, chunki u matnni faqat bir marta ko'rib chiqadi, bu esa uning samaradorligini sezilarli darajada oshiradi.

KMP Algoritmining Asosiy Prinsipi

KMP algoritmi asosan ikki bosqichga bo'linadi: preprocessing va qidiruv bosqichi. Ushbu ikkita bosqich matnni qidirish jarayonini samarali bajarish uchun bir-birini to'ldiradi.

Preprocessing Bosqichi: Bu bosqichda matnning prefiks va sufiks xususiyatlari aniqlanadi. Har bir belgi o'zidan oldingi belgilarga qanchalik mos kelishini tahlil qiladigan lps (longest prefix-suffix) massivi quriladi. Bu massiv matnni qidirish jarayonida kerakli ma'lumotlarni saqlaydi va kelajakdagi qidiruv operatsiyalarini optimallashtiradi.

Qidiruv Bosqichi: Preprocessing bosqichida olingan lps massivi yordamida matnni qidirish jarayoni amalga oshiriladi. Agar biror belgilar mos kelmasa, u holda algoritm shunchaki qidiruvni davom ettiradi, lekin lps massividan foydalanib, takrorlanadigan qadamlarni cheklaydi va matnni qayta ko'rib chiqmasdan keyingi belgiga o'tadi.

KMP algoritmi qidiruv jarayonida matnning har bir belgisini faqat bir marta tekshiradi. Bu esa uni $O(n + m)$ vaqt murakkabligiga ega qiladi, bu yerda n — matn uzunligi, m esa submatn uzunligi.

KMP Algoritmining Afzalliklari

Tezlik va samaradorlik: KMP algoritmi o'zining samaradorligi bilan tanilgan, chunki u qidiruv jarayonida har bir belgi bilan faqat bir marta ishlaydi. Bu algoritmni boshqa an'anaviy qidiruv algoritmlari bilan taqqoslaganda, masalan, Brute-force algoritmi bilan solishtirganda, sezilarli tezlikka ega bo'ladi.

Takrorlanadigan operatsiyalarni minimallashtirish: KMP algoritmi, o'zining preprocessing bosqichi orqali, matnni qayta ko'rib chiqmasdan, mavjud ma'lumotlar yordamida qidiruv jarayonini davom ettiradi. Bu xususiyat algoritmni juda samarali qiladi, chunki ortiqcha hisoblashlar va takrorlanadigan ishlar yo'q.





MODERN PROBLEMS IN EDUCATION AND THEIR SCIENTIFIC SOLUTIONS

Katta matnlar bilan ishlash: KMP algoritmi katta matnlar bilan ishlashda juda foydalidir, chunki uning ishlash vaqti matn uzunligiga bevosita bog'liq. Bu unga real vaqt tizimlarida ham samarali ishlash imkoniyatini beradi.

KMP Algoritmi Qayerda Qo'llaniladi?

KMP algoritmi, o'zining yuqori samaradorligi va tezligi bilan, turli sohalarda keng qo'llaniladi:

Matnni qidirish dasturlari: KMP algoritmi matn qidiruv tizimlarida, masalan, fayl tizimlarida yoki matnli hujjatlarni izlashda ishlatiladi. Ushbu algoritm matndagi submatnni tez va samarali topish uchun ishlatiladi.

Bioinformatika: Genetik tahlil va genomni qidirishda KMP algoritmi, DNK ketma-ketliklarini tez topish uchun ishlatiladi. Bioinformatika sohasida bu algoritmni qo'llash ilmiy tadqiqotlarni samarali o'tkazish uchun katta ahamiyatga ega.

Versiya boshqaruvi tizimlari: Kod tahlili va versiyalarni solishtirishda KMP algoritmi keng qo'llaniladi. Dasturchilar yangi va eski kod o'rtasidagi farqlarni topishda bu algoritmni ishlatadilar.

Siqish algoritmlari: Ba'zi siqish algoritmlarida matnni optimallashtirish va submatnlarni aniqlash uchun KMP algoritmi ishlatiladi. Bu algoritm ma'lum bir ketma-ketliklarni aniqlash va takrorlanuvchi qismlarni saqlab qolish imkonini beradi.

KMP Algoritmi vs Boshqa Algoritmlar

KMP algoritmi Brute-force algoritmiga qaraganda ancha samarali hisoblanadi. Brute-force algoritmi har bir pozitsiyani tekshirib chiqadi, bu esa yirik matnlar bilan ishlaganda juda sekinlashishi mumkin. KMP esa faqat bir marta matnni ko'rib chiqadi, shuning uchun undan ancha tezroq ishlaydi. Shuningdek, Boyer-Moore algoritmi ham matn qidirishda samarali bo'lsa-da, u KMP algoritmiga nisbatan murakkabroq va ba'zi hollarda kamroq samarali bo'lishi mumkin.

Masala: Foydalanuvchi matn va patternlar ro'yxatini kiritadi. Dastur har bir patternni matnda izlaydi va topilgan indekslarni ekranga chiqaradi.

Yechim:

Bu masalani yechishda quyidagi bosqichlarni amalga oshiramiz:

1. Patternlar ro'yxatini olish.
2. Har bir pattern uchun prefiks-funksiyani hisoblash.
3. Matnda har bir patternni qidirish va ularning topilgan indekslarini chiqarish.

C# Dasturi:

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

```
class KMPAlgorithm  
{  
    // Prefiks-funksiyani hisoblash
```



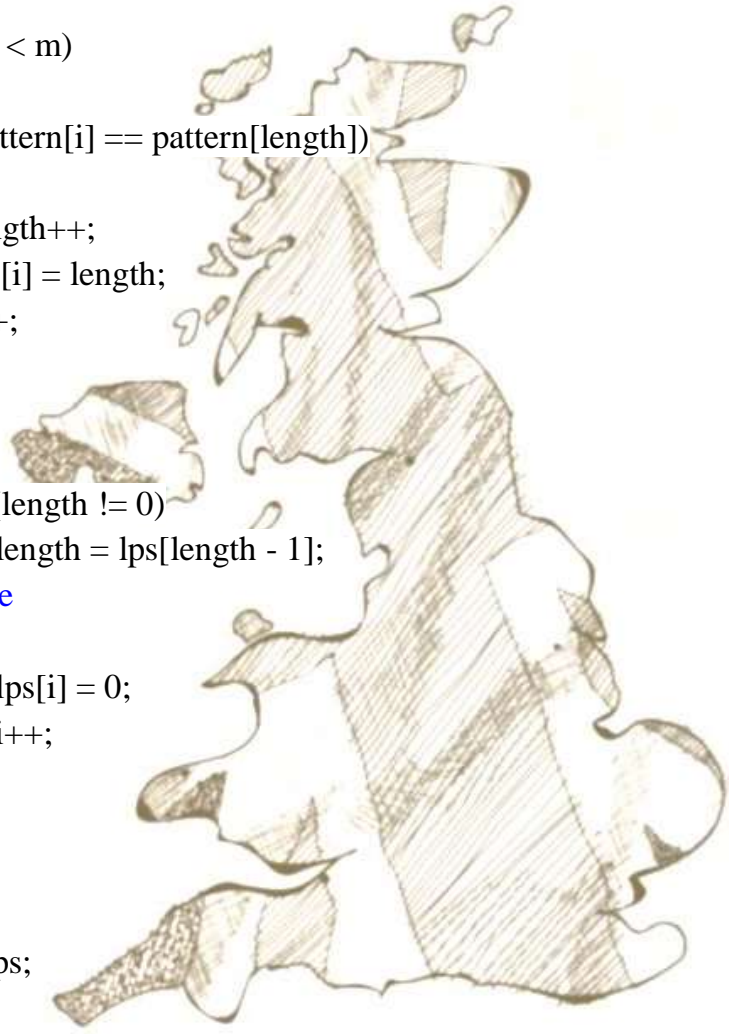


MODERN PROBLEMS IN EDUCATION AND THEIR SCIENTIFIC SOLUTIONS

```
static List<int> ComputeLPSArray(string pattern)
{
    int m = pattern.Length;
    List<int> lps = new List<int>(new int[m]);
    int length = 0;
    int i = 1;

    while (i < m)
    {
        if (pattern[i] == pattern[length])
        {
            length++;
            lps[i] = length;
            i++;
        }
        else
        {
            if (length != 0)
                length = lps[length - 1];
            else
            {
                lps[i] = 0;
                i++;
            }
        }
    }

    return lps;
}
```



// KMP algoritmi bo'yicha qidirish

```
static void KMPSearch(string text, string pattern)
{
    int n = text.Length;
    int m = pattern.Length;

    List<int> lps = ComputeLPSArray(pattern);
    int i = 0; // index for text
    int j = 0; // index for pattern
}
```





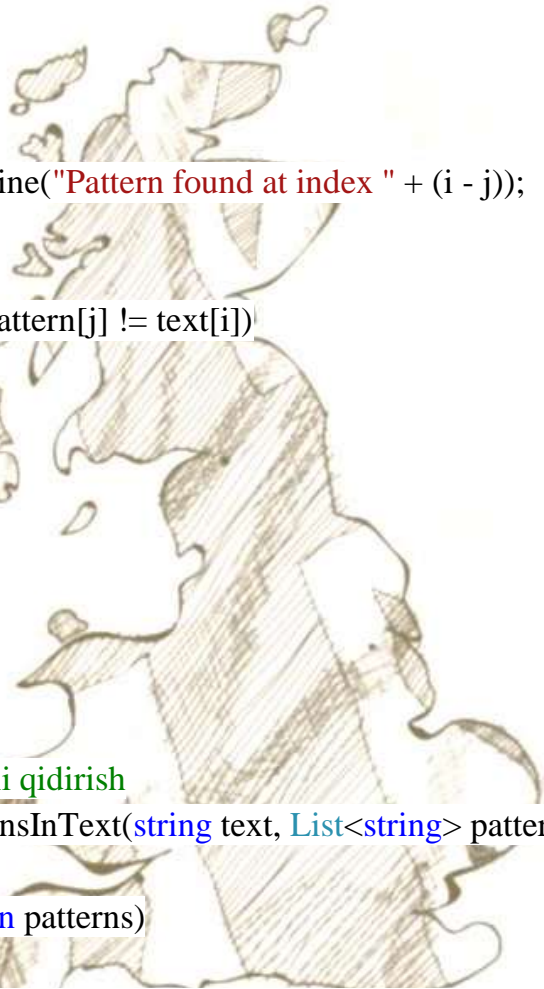
MODERN PROBLEMS IN EDUCATION AND THEIR SCIENTIFIC SOLUTIONS

```
while (i < n)
{
    if (pattern[j] == text[i])
    {
        i++;
        j++;
    }

    if (j == m)
    {
        Console.WriteLine("Pattern found at index " + (i - j));
        j = lps[j - 1];
    }
    else if (i < n && pattern[j] != text[i])
    {
        if (j != 0)
            j = lps[j - 1];
        else
            i++;
    }
}

// Bir nechta patternlarni qidirish
static void SearchPatternsInText(string text, List<string> patterns)
{
    foreach (var pattern in patterns)
    {
        Console.WriteLine($"Searching for pattern: {pattern}");
        KMPSearch(text, pattern);
        Console.WriteLine(); // Bo'sh qator
    }
}

// Dastur boshqaruvi
static void Main(string[] args)
{
    Console.WriteLine("Enter the text:");
    string text = Console.ReadLine();
}
```





MODERN PROBLEMS IN EDUCATION AND THEIR SCIENTIFIC SOLUTIONS

```
Console.WriteLine("Enter number of patterns to search:");  
int n = int.Parse(Console.ReadLine());
```

```
List<string> patterns = new List<string>();  
for (int i = 0; i < n; i++)  
{  
    Console.WriteLine($"Enter pattern {i + 1}:");  
    patterns.Add(Console.ReadLine());  
}  
  
// Patternlarni matnda qidirish  
SearchPatternsInText(text, patterns);  
}  
}
```

Dastur qanday ishlaydi:

1. ComputeLPSArray funksiyasi — pattern uchun prefiks-funksiyasini hisoblaydi.
2. KMPSearch funksiyasi — matnda patternni KMP algoritmi yordamida qidiradi.
3. SearchPatternsInText funksiyasi — bir nechta patternlarni matnda izlaydi va har birini qidiradi.
4. Main funksiyasi — foydalanuvchidan matn va patternlar ro'yxatini olib, ularni izlashni boshlaydi.

Misol uchun:

Foydalanuvchi quyidagilarni kiritishi mumkin:

Matn: "ABABDABACDABABCABAB"

Patternlar:

- "ABAB"
- "ABC"
- "DAB"

Natija

Enter the text:

ABABDABACDABABCABAB

Enter number of patterns to search:

3

Enter pattern 1:

ABAB

Searching for pattern: ABAB

Pattern found at index 0

Pattern found at index 10

Pattern found at index 15





MODERN PROBLEMS IN EDUCATION AND THEIR SCIENTIFIC SOLUTIONS

Enter pattern 2:

ABC

Searching for pattern: ABC

Pattern found at index 12

Enter pattern 3:

DAB

Searching for pattern: DAB

Pattern found at index 4

Knut-Morris-Pratt (KMP) algoritmi string matching, ya'ni matnda substring qidirish muammolarini juda samarali hal qilishda qo'llaniladigan algoritmdir. An'anaviy string qidirish algoritmlarida substringni izlashda har bir belgi ketma-ket solishtiriladi, natijada takroriy qidirishlar amalga oshiriladi. KMP algoritmi esa bu jarayonda ortiqcha solishtirishlardan qochadi va prefiks-funksiya yordamida qidiruvni tezlashtiradi. Prefiks-funksiya substringning o'zidagi prefiks va suffikslarning bir-biriga mos kelish darajasini hisoblaydi, shuning uchun qidiruv jarayonini optimallashtirib, umumiy murakkablikni kamaytiradi.

KMP algoritmi $O(n + m)$ murakkablikda ishlaydi, bu esa juda katta hajmdagi matnlarda ham izlash jarayonini sezilarli darajada tezlashtiradi. Bundan tashqari, u bir qator sohalarda, jumladan, qidiruv tizimlari, genetik tahlil, va keng ko'lamdagi matnlarni tahlil qilishda keng qo'llaniladi. Algoritmning afzalligi — takroriy qidiruvlardan qochish va qiyin holatlarda ham samaradorlikni saqlab qolishidir. Umuman olganda, KMP algoritmi string matching muammolariga nisbatan samarali va ishonchli yechim bo'lib, uning ko'plab amaliy qo'llanmalari mavjud. Algoritmning kelajakda string tahlili va qidiruv sohasidagi o'rni yanada mustahkamlanishi kutilmoqda, chunki uning ishlashi tezligi va resurslardan samarali foydalanishi zamonaviy texnologiyalar uchun katta ahamiyatga ega.

FOYDALANGAN ADABIYOTLAR:

1. Knut, D. E., Morris, J. H., & Pratt, V. R. (1977). "Fast pattern matching in strings." *SIAM Journal on Computing*, 6(2), 323-350.
2. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009).
3. Gusfield, D. (1997). *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*.
4. Skiena, S. S. (2008). *The Algorithm Design Manual* (2nd ed.).
5. GeeksforGeeks. "KMP Algorithm for Pattern Searching."
6. Rosen, K. H. (2012). *Discrete Mathematics and Its Applications* (7th ed.). McGraw-Hill Education.





MODERN PROBLEMS IN EDUCATION AND THEIR SCIENTIFIC SOLUTIONS

7. Levitin, A. (2011). *Introduction to the Design and Analysis of Algorithms* (3rd ed.). Pearson.
8. Farmonov, S. R. (2024). BFS ALGORITMI ORQALI TOPOLOGIK TARTIBNI ANIQLASH. ОБРАЗОВАНИЕ И НАУКА В XXI BEKE, (56-5).
9. Raxmonjonovich, F. S. (2024). JONSON ALGORITMI BILAN BOG ‘LIQ MUAMMOLAR VA ULARNI YECHISH USULLARI. *Modern education and development*, 15(4), 32-42.
10. Raxmonjonovich, F. S. (2024). KAN ALGORITMI ASOSIDA GRAFLARDA SIKLSIZ TARTIBNI ANIQLASH. *Modern education and development*, 15(4), 43-49.
11. Rahmonaliyevich, F. S. (2024). FORD-BELLMAN ALGORITMI, ENG QISQA YO ‘LLARNI TOPIH. TADQIQOTLAR. UZ, 51(2), 45-51.
12. 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).
13. Farmonov, S., & Ergashaliyeva, B. (2024). QAT'YMAS NEYRON TO'RLAR: MAMDANI QAT'YMAS MANTIQIY XULOSASI, SUGENO QAT'YMAS MANTIQIY XULOSASI. *Development and innovations in science*, 3(5), 62-70.
14. Raxmonjonovich, F. S. (2024). KOMPYUTER GRAFIKASI VA O‘YIN DASTURLASHDA JOHNSON ALGORITMINING AHAMIYATI. *Ta'lim innovatsiyasi va integratsiyasi*, 34(2), 145-151.
15. Farmonov, S. R., & qizi Oktamjonova, M. I. (2024, November). FLOYD–UORSHELL ALGORITMI. In *International Conference on World Science and Resarch* (Vol. 1, No. 3, pp. 32-42).
16. Raxmonjonovich, F. S. (2024). XOFMAN KODLASH TIZIMI: AVIATSIYA VA PARVOZ MA'LUMOTLARINI SIQISHNING INNOVATSION YONDASHUVI. *Лучшие интеллектуальные исследования*, 34(1), 153-160.
17. Raxmonjonovich, F. S., & Botirali o‘g‘li, T. M. (2024). KAN ALGORITMINI GRAFLARDA QO‘LLANILISHI. TADQIQOTLAR. UZ, 51(2), 27-36.
18. Raxmonjonovich, F. S. (2024). ROBOTOTEXNIKA SOHASIDA GEOMETRIK ALGORITMLARNING O‘RNI. *Лучшие интеллектуальные исследования*, 34(1), 134-141.
19. Raxmonjonovich, F. S. (2024). MINIMAL BOG‘LANISH DARAXTINI TOPISHDA PRIM ALGORITMIDAN FOYDALANISH. YANGI O‘ZBEKISTON, YANGI TADQIQOTLAR JURNALI, 1(3), 436-443.

