

MA'LUMOTLAR BAZASINI OPTIMALLASHTIRISHDA RABIN-KARP
ALGORITMINING QO'LLANILISHI

Farmonov Sherzodbek Raxmonjonovich

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

[*farmonovsh@gmail.com*](mailto:farmonovsh@gmail.com)

Ikromaliyeva Oynisa Islomjon qizi

Farg'ona davlat universiteti 2-kurs talabasi

[*oynisaikromaliyeva050@gmail.com*](mailto:oynisaikromaliyeva050@gmail.com)

Annotatsiya. Mazkur maqolada ma'lumotlar bazasini optimallashtirishda Rabin-Karp algoritmining qo'llanilishi yoritiladi. Rabin-Karp algoritmi stringlarni qidirishda samarali va oddiy usul sifatida tanilgan bo'lib, uning asosiy prinsipi hashing texnikasiga tayanadi. Maqolada algoritmning ishlash jarayoni, xususan, substringlarni tezkor qidirish va moslikni tekshirish uchun hashingdan foydalanish jarayoni batafsil bayon qilinadi. Shuningdek, algoritmni katta hajmdagi ma'lumotlar bazasida qo'llashning afzalliklari, masalan, qidiruv vaqtini kamaytirish va resurslardan samarali foydalanish bo'yicha tahlillar keltiriladi

Kalit so'zlar: Rabin-Karp algoritmi, Hashlash texnikasi, Substring qidirish, Algoritmning murakkabligi, Plagiatni aniqlash, ketma-ketlikni solishtirish.

Annotation. В данной статье рассматривается применение алгоритма Рабина-Карпа для оптимизации баз данных. Алгоритм Рабина-Карпа известен как эффективный и простой метод поиска строк, основным принципом которого является использование хеширования. В статье подробно описан процесс работы алгоритма, в частности, использование хеширования для быстрого поиска подстрок и проверки совпадений. Кроме того, представлены преимущества применения алгоритма в больших базах данных, такие как сокращение времени поиска и эффективное использование ресурсов.

Ключевые слова: алгоритм Рабина-Карпа, техника хеширования, поиск подстрок, сложность алгоритма, обнаружение плагиата, сравнение последовательностей.

Аннотация. This article discusses the application of the Rabin-Karp algorithm in database optimization. The Rabin-Karp algorithm is known as an efficient and straightforward method for string searching, with its core principle being the use of hashing. The article provides a detailed explanation of how the algorithm works, particularly its use of hashing for fast substring searching and matching verification. Additionally, the advantages of applying the algorithm in large-scale databases, such as reducing search time and efficiently utilizing resources, are highlighted.

Keywords: Rabin-Karp algorithm, hashing technique, substring search, algorithm complexity, plagiarism detection, sequence comparison.

Rabin-Karp algoritmi-matn ichidan substringni izlash uchun ishlatiladigan algoritmi. U hashlash texnikasidan foydalanadi.

- Hashlash texnikasi-matnni qidirishni tezlashtirish uchun substringlarni qiymatga aylantirish jarayoni.

- Substring qidirish-matnning bir qismini boshqa bir matnda topish jarayoni.

- Algoritmning murakkabligi-Rabin-Karp algoritmining eng yaxshi holatda $O(n+m)$ va eng yomon holatda $O(n*m)$ bo'lgan ishlash samaradorligi.

- Plagiatni aniqlash-Rabin-Karp algoritmi turli hujjatlar yoki matnlar orasidagi o'xshash qismlarni topishda ishlatiladi

- Ketma-ketliklarni soddalashtirish-matn yoki ma'lumotlar ketma-ketliklarini taqqoslash jarayoni, algoritmning asosiy qo'llanilish sohalaridan biri.

Rabin-Karp algoritmi. Rabin-Karp algoritmi 1987-yilda Maykl Rabin va Richard Karp tomonidan ishlab chiqilgan. Algoritm substring qidirish vazifasini hash yuqori samaradorlikka ega bo'lib, ayniqsa, bir vaqtning o'zida bir nechta substringlarni qidirishda juda samaralidir. Algoritmning ishlash asosiy bosqichlari quyidagicha:

- 1-bosqich. Qidirilayotgan substringning hash qiymati hisoblanadi.

- 2-bosqich. Matnda ketma-ket substringlarning hash qiymatlari hisoblanib, ular qidirilayotgan substringning hash qiymati bilan solishtiriladi.

- 3-bosqich. Agar hash qiymatlari mos kelsa, hash to'qnashuvlarni aniqlash uchun substringlar belgi-belgi solishtiriladi.

Ma'lumotlar bazasini optimallashtirishda Rabin-Karp algoritmining ahamiyati quyidagilarda iborat:

- **Takroriy yozuvlarni aniqlash va bartaraf etish.**

Ma'lumotlar bazasida takroriy yozuvlar ishlash samaradorligini pasaytiradi va keraksiz xotira egallaydi. Rabin-Karp algoritmi substring hash qiymatlarini tezkor taqqoslash orqali takroriy yozuvlarni samarali aniqlash imkonini beradi. Masalan:

1. Katta hajmdagi matnli qatorlarda o'xshash qismlarni topish.

2. Takrorlanayotgan ma'lumotlarni birlashtirish yoki olib tashlash

- **Plagiat va o'xshashlikni aniqlash.**

Matnli ma'lumotlar bazasida hujjatlar, maqolalar yoki boshqa ma'lumotlarning o'xshash qismlarini aniqlash uchun Rabin-Karp algoritmi samarali qo'llaniladi. Bu jarayon quyidagicha amalga oshiriladi:

1. Har bir hujjatdagi substringlarning hash qiymatlari hisoblanadi.

2. Hujjatlar o'zaro taqqoslanib, o'xshash substringlar aniqlanadi.

Rabin-Karp algoritmining afzalliklari va cheklovlari:

•**Afzalligi:Hashing yordamida tezkor qidiruv.**Algoritm hashing texnikasiga asoslanganligi sababli substringlarni qidirishda samaradorlikni oshiradi. Katta hajmdagi ma'lumotlar ichidan mos yozuvlarni topishda vaqtni tejaydi.

•**Ko'p qidiruv shablonlari bilan ishlash.**Rabin-Karp algoritmi bir vaqtning o'zida bir nechta qidiruv shablonlarini ishlatishi mumkin, bu esa ma'lumotlar bazasida murakkab qidiruv talab qilinadigan vaziyatlarda foydalidir.

•**Cheklovlari:Hash to'qnashuvlari (Collisions).**Hash funksiyalari to'qnashuvlarni yuzaga keltirishi mumkin, ya'ni turli substringlar bir xil hash qiymatini oladi. Bu esa algoritmnining samaradorligini pasaytirishi va qo'shimcha tekshiruv talab qilishi mumkin.

•**Hashingni hisoblashning xarajati.**Katta hajmdagi yozuvlar uchun har bir substringning hash qiymatini hisoblash qimmatga tushishi mumkin, ayniqsa, uzoq ketma-ketliklar bilan ishlaganda.

•**Moslikni tekshirish uchun qo'shimcha vaqt.**Hash qiymatlari mos kelsa substringlarni qayta-qayta taqqoslash talab qilinadi, bu esa algoritmnining ishlash tezligini pasaytiradi.

Masala:Bir kompaniya katta hajmdagi matnli ma'lumotlarni saqlovchi ma'lumotlar bazasidan foydalanuvchilarning qidiruv talablarini tezkorlik bilan amalga oshirishni istaydi. Kompaniyaning talablariga ko'ra, har bir yozuvda foydalanuvchi so'ragan kalit so'zning mavjudligini aniqlash kerak. Shu bilan birga, qidiruvning umumiy vaqti ma'lumotlar bazasining hajmi kattalashgan taqdirda ham sezilarli darajada oshib ketmasligi lozim.

Shartlar:

1. Ma'lumotlar bazasi tarkibi:

Ma'lumotlar bazasida quyidagi matnlar saqlangan:

"Dasturlash algoritmlari va ma'lumotlar bazasi dizayni"

"Rabin-Karp algoritmi yordamida samarali qidiruv"

"Ma'lumotlar bazasida optimallashtirish va indekslash"

"Algoritmlar va ma'lumotlar bazalari bo'yicha tadqiqotlar"

2. Kalit so'zi:"algoritm" so'zi qaysi matnlar ichida uchrashini bilishi kerak.

3. Cheklovlar va talablar:Ma'lumotlar bazasi 10,000 ta yoki undan ko'p yozuvlardan iborat bo'lishi mumkin.Algoritm matnni hashlash orqali qidirishni amalga oshirishi lozim.Qidiruv natijalarini qaytarish vaqti 1 sekunddan oshmasligi lozim.

C# dagi kodi

```
using System;
```

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

```
class Program
```

```
{// Rabin-Karp algoritmiga asoslangan substring qidiruv funksiyasi
```

```
public static List<int> RabinKarpSearch(string text, string pattern, int prime =
```

101)

```

{
int textLength = text.Length;
int patternLength = pattern.Length;
int patternHash = 0; // Kalit so'zning hash qiymati
int textHash = 0; // Matnning joriy oynasi uchun hash qiymati
int hashBase= 256; // Hash uchun asos (ASCII qiymatlari uchun)
int highestPow = 1; // Eng yuqori darajali koeffitsient
List<int> a = new List<int>();
// Eng yuqori darajali koeffitsientni hisoblash (hashAsos^(kalitsozLength-1) %
prime)
for(int i = 0; i < patternLength - 1; i++)
highestPow = (highestPow * hashBase) % prime;
// Boshlang'ich hash qiymatlarini hisoblash
for (int i = 0; i < patternLength; i++)
{
patternHash = (hashBase* patternHash + pattern[i]) % prime;
textHash = (hashBase * textHash + text[i]) % prime;
} // Matn ustida harakat qilish
for(int i = 0; i <= textLength - patternLength; i++)
{// Agar hash qiymatlari teng bo'lsa, to'liq solishtirish
if (patternHash == textHash)
{
bool match = true;
for (int j = 0; j < patternLength; j++)
{
if (text[i + j] != pattern[j])
{
match = false; break;
}
}
if (match)
a.Add(i); // Topilgan indeksni ro'yxatga qo'shish
} // Keyingi oynaning hash qiymatini hisoblash
if (i < textLength - patternLength)
{
textHash = (hashBase * (textHash - text[i] * highestPow) + text[i +
patternLength]) % prime;
// Manfiy hash qiymatlarini musbatga o'tkazish

```


kalit so'zni qidirish uchun RabinKarpSearch funksiyasidan foydalanadi. Qidiruv natijasida kalit so'z mavjud yoki yo'qligini ko'rsatadi.

Rabin-Karp algoritmi stringlarni qidirish va moslikni tekshirish bo'yicha samarali va sodda usuldir. Uning asosiy prinsipi substringlarning hash qiymatini hisoblash orqali tezkor qidiruvni amalga oshirishga asoslangan. Algoritm bir vaqtning o'zida bir nechta shablon bilan ishlash, katta hajmdagi matnlarni qidirish va ma'lumotlarni tahlil qilishda yuqori samaradorlik ko'rsatadi. Rabin-Karp algoritmi ma'lumotlar bazasidagi takroriy yozuvlarni aniqlash va optimallashtirish jarayonida ishlatilishi mumkin.

Foydalanilgan adabiyotlar:

1. Richard M. Karp, Michael O. Rabin. "Efficient Randomized Pattern-Matching Algorithms." IBM Journal of Research and Development, 1987.
2. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. "Introduction to Algorithms." MIT Press, 3rd Edition, 2009.
3. Jeffrey D. Ullman, Jennifer Widom. "A First Course in Database Systems." Pearson, 3rd Edition, 2008.
4. Elmasri, R., Navathe, S. B. "Fundamentals of Database Systems." Pearson, 7th Edition, 2015.
5. Gusfield, Dan. "Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology." Cambridge University Press, 1997.
6. Peter Norvig. "Paradigms of Artificial Intelligence Programming: Case Studies in Common Lisp." Morgan Kaufmann, 1992.
7. Knuth, D. E., Morris, J. H., Pratt, V. R. "Fast Pattern Matching in Strings." SIAM Journal on Computing, 1977.
8. Gonzalez, T. F. "Handbook of Approximation Algorithms and Metaheuristics." Chapman & Hall/CRC, 2007.
9. Aho, A. V., Hopcroft, J. E., & Ullman, J. D. (1974). The Design and Analysis of Computer Algorithms. Addison-Wesley.
10. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to Algorithms (3rd Edition). MIT Press.
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.