

Knut-Morris-Prat algoritmi

Onarkulov Maksadjon Karimberdiyevich

Farg'ona davlat universiteti Amaliy matematika va
informatika kafedrası dotsenti (PhD)

maxmaqsad@gmail.com

Habibullayev Javohir Odilbek o'g'li

Farg'ona davlat universiteti 2-bosqich talabasi

habibullayevjavohirbek5@gmail.com

Anatatsiya:

Mazkur maqolada matn ichida namunaviy qatorni (substring) qidirishda samarali ishlovchi Knut-Morris-Pratt (KMP) algoritmi tahlil qilinadi. Ushbu algoritmi klassik qidiruv algoritmlaridan farqli ravishda takroriy taqqoslashlarni kamaytiradi va $O(n)$ vaqt murakkabligiga ega. KMP algoritmi matn tahlili, bioinformatika, qidiruv tizimlari va plagiat aniqlash kabi sohalarda keng qo'llaniladi. Maqolada algoritmnining ishlash prinsipi, prefiks-funksiyani qurish bosqichlari va C# tilida dasturlashdagi amaliy namunalar keltirilgan.

Kalit so'zlar: Knut-Morris-Pratt algoritmi, qidiruv algoritmlari, substring qidiruvi, prefiks-funksiyasi, algoritmi murakkabligi, C# dasturlash, matnni qayta ishlash

Anatation:

This paper analyzes the Knuth-Morris-Pratt (KMP) algorithm, which is an efficient method for searching substrings within a given text. Unlike traditional string search algorithms, KMP avoids redundant comparisons by utilizing a preprocessed prefix function, achieving a time complexity of $O(n)$. The algorithm is widely used in fields such as text processing, bioinformatics, search engines, and plagiarism detection. The paper explores the core principles of KMP, details the construction of the prefix function, and provides practical examples implemented in C#.

Keywords: Knuth-Morris-Pratt algorithm, pattern matching, substring search, prefix function, algorithm complexity, C# programming, text processing

Аннотация:

В данной статье рассматривается алгоритм Кнута-Морриса-Пратта (КМП), являющийся эффективным методом поиска подстроки в тексте. В отличие от традиционных алгоритмов, КМП минимизирует количество ненужных сравнений за счёт использования заранее вычисленной префикс-функции, обеспечивая временную сложность $O(n)$. Алгоритм широко используется в обработке текстов, биоинформатике, поисковых системах и системах обнаружения плагиата. В статье подробно изложен принцип работы алгоритма, этапы построения префикс-функции и приведены практические примеры на языке программирования C#.

Ключевые слова: Алгоритм Кнута-Морриса-Пратта, поиск шаблонов, поиск подстроки, префикс-функция, сложность алгоритма, программирование на C#, обработка текста

Matn ichida ma'lum bir namunani yoki so'zni qidirish masalasi kompyuter fanining eng asosiy va keng tarqalgan muammolaridan biridir. Har qanday matnli ma'lumotlar bazasida, hujjatlarni qayta ishlashda, veb-saytlarda izlash tizimlarida yoki dasturlash tillarida kod qidirishda samarali va tezkor qidiruv algoritmlari talab qilinadi. Oddiy, ya'ni naive algoritmi har bir pozitsiyadan boshlab satrning har bir belgisi bilan namunani taqqoslash orqali ishlaydi, ammo bu yondashuv katta hajmdagi matnlarda sekin va samarasiz bo'lib qoladi. Ayniqsa, uzun matn va qidirilayotgan namunada ko'p takrorlanuvchi belgilar mavjud bo'lsa, oddiy yondashuv ko'plab keraksiz taqqoslashlarga sabab bo'ladi. Shu nuqtai nazardan, samarali va optimallashtirilgan algoritmlarga ehtiyoj tug'iladi. Knut-Morris-Pratt (KMP) algoritmi aynan shunday masalani hal qilish uchun ishlab chiqilgan. U prefiks-funksiyasi asosida ishlaydi va keraksiz taqqoslashlarni minimallashtirish orqali qidiruv jarayonini sezilarli darajada tezlashtiradi. Bu algoritmning samaradorligi va ishonchliligi ko'plab amaliy sohalarda, jumladan, matn tahlili, biologik ketma-ketliklarni o'rganish, plagiat aniqlash va real

vaqt tizimlarida keng qo'llanilishiga olib keldi. Ushbu maqolada KMP algoritmining nazariy asoslari, ishlash printsiplari va uning amaliy ahamiyati batafsil yoritiladi, shuningdek, boshqa matn qidiruv algoritmlari bilan solishtirish orqali uning o'ziga xos afzalliklari ko'rsatib beriladi. Maqolaning maqsadi o'quvchilarga KMP algoritmini chuqur tushunish va uni samarali qo'llash bo'yicha bilim berishdir.

Knut-Morris-Pratt (KMP) algoritmi satrlar ichida namunani (pattern) samarali izlash uchun ishlab chiqilgan algoritmdir. Uning asosiy afzalligi — har bir belgini faqat bir marta ko'rib chiqishi, ya'ni ortiqcha taqqoslashlarni oldini olishi bilan bog'liq. KMP algoritmi ikki asosiy bosqichdan iborat: birinchi bosqichda qidirilayotgan namunaga tegishli maxsus yordamchi massiv — prefiks-funksiyasi (yoki failure function) quriladi. Ushbu prefiks-funksiyada har bir indeksda o'sha pozitsiyagacha bo'lgan substring ichida qanday qilib o'zi bilan bir xil boshlanadigan va tugaydigan eng uzun substring mavjudligi haqida ma'lumot saqlanadi. Bu yordamchi massiv kelajakdagi qidiruvda qaysi belgidan davom etishni aniqlashda muhim rol o'ynaydi.

Algoritmning ikkinchi bosqichi — to'g'ridan-to'g'ri matn ichida namunani izlash jarayoni hisoblanadi. Bu bosqichda matn va namunadagi belgilar navbatma-navbat taqqoslanadi. Belgilar bir-biriga mos kelgan holatda ikkala indeks oldinga siljutiladi. Agar mos kelmaslik aniqlansa, KMP algoritmi prefiks-funksiyadagi qiymatlardan foydalanib, namunadagi indeksni shunday o'zgartiradiki, u ilgari tekshirib chiqilgan mos kelgan belgilarni qayta ko'rib chiqmaydi. Bu yondashuv natijasida har bir belgining maksimal bir marta tahlil qilinishi ta'minlanadi, bu esa umumiy ishlash tezligini $O(n + m)$ — ya'ni matn uzunligi n va namunalar uzunligi m ga bog'liq bo'lgan chiziqli murakkablikda saqlab qoladi.

KMP algoritmining ishlash mantig'i zahirada optimallik tamoyili yotadi. Oddiy string qidiruv algoritmlarida biror mos kelmaslik aniqlangach, butun namunani qaytadan taqqoslashga to'g'ri keladi. KMP esa shu bosqichgacha o'zaro mos kelgan belgilarni tahlil qilganligi tufayli, mos kelmaslik holida to'liq qaytish emas, balki faqat zarur pozitsiyadan davom etishni ta'minlaydi. Bu esa algoritmning samaradorligini keskin oshiradi. KMP algoritmi nafaqat nazariy jihatdan, balki amaliyotda — masalan,

matnli fayllarda ma'lumot izlash, plagiat aniqlash, kodni indekslash kabi vazifalarda ham keng qo'llaniladi.

Knut-Morris-Pratt (KMP) algoritmining amaliy qo'llanilishi juda keng va u turli sohalarda samarali ishlatiladi. Eng avvalo, matnni tahlil qilish bilan bog'liq vazifalarda — masalan, fayllarda yoki hujjatlarda aniq bir ifodani izlashda KMP algoritmi ancha yuqori samaradorlikka ega. Katta hajmdagi matnlar ustida ishlaydigan dasturlar, masalan, fayl qidiruv tizimlari, kod indeksatsiyasi yoki oddiy matn muharrirlari (masalan, Notepad++, VS Code) ushbu algoritmdan foydalanadi. U yerda foydalanuvchi izlayotgan so'z yoki ibora minglab sahifalar ichida tez va ishonchli tarzda aniqlanadi.

KMP algoritmi, shuningdek, biologik ma'lumotlar tahlilida ham keng qo'llaniladi. Masalan, DNK yoki RNK ketma-ketliklarida ma'lum bir genetik andozani (sequence pattern) aniqlashda bu algoritm samarali vosita sifatida ishlaydi. Genom tadqiqotlarida ma'lum bir fragmentning turli organizmlardagi mavjudligini aniqlash zarur bo'lsa, KMP algoritmi yirik genom fayllari ichida ushbu ketma-ketlikni tezlik bilan topadi.

Bundan tashqari, KMP algoritmi xavfsizlik tizimlarida ham qo'llaniladi. Masalan, dasturiy ta'minotlar yoki tizim loglari ichida zararli so'zlar yoki foydalanuvchi faoliyatidagi nojo'ya harakatlarni aniqlash uchun filtr tizimlari matnlar ichida aniqlovchi izlash mexanizmidan foydalanadi. Shu kabi holatlarda KMP algoritmi yuqori aniqlik va tezlik bilan ishlaydi. Shu bilan birga, plagiat aniqlash tizimlarida ham matnlarni o'zaro taqqoslash orqali bir-biriga o'xshash segmentlarni aniqlashda samarali bo'lib, bir matn ichidan boshqa matn bilan qanchalik o'xshashlik borligini topishga yordam beradi.

Dasturlash tillarida KMP algoritmi C#, Java, Python kabi tillarda osonlik bilan amalga oshiriladi. Turli standart kutubxonalar orqali yoki o'zi yozilgan funksiyalar yordamida matn ichida izlashni amalga oshirish mumkin. Dasturchilar bu algoritmnini o'z ilovalariga integratsiya qilish orqali foydalanuvchilarga tezkor va aniq qidiruv imkoniyatini yaratadilar. Shu bilan birga, KMP algoritmi real vaqtda ishlaydigan

tizimlarda, masalan, chat ilovalari yoki matnni avtomatik tahlil qiluvchi vositalarda ham qo'llaniladi. Bu uning universalligini va amaliy qiymatini yana bir bor tasdiqlaydi.

Knut-Morris-Pratt (KMP) algoritmini boshqa matn qidiruv algoritmlari bilan solishtirganda, uning samaradorligi va ishlash prinsipidagi farqlar yaqqol namoyon bo'ladi. Eng oddiy algoritm — bu “naive” yoki to'g'ridan-to'g'ri izlash algoritmidir. Unda har bir pozitsiyadan boshlab namunani matn bilan taqqoslash amalga oshiriladi va mos kelmaslik aniqlansa, keyingi belgidan davom etiladi. Bu yondashuvda eng yomon holatda algoritmning vaqt murakkabligi $O(n \cdot m)$ bo'lib, bu esa katta matnlar va uzun namunalar uchun sekin ishlashga olib keladi. KMP algoritmi esa prefiks-funksiyasi yordamida ortiqcha taqqoslashlarni cheklaydi va umumiy murakkablikni $O(n + m)$ ga tushiradi. Bu esa uni samaraliroq va real vaqt tizimlari uchun mosroq qiladi.

KMP algoritmi Boyer-Moore algoritmi bilan ham ko'p taqqoslanadi. Boyer-Moore algoritmi matnni o'ngdan chapga qarab tahlil qilishi va bir nechta oldindan hisoblangan qoidalar (bad character va good suffix) asosida belgilarning tahlilini o'tkazishi bilan mashhur. U amalda ko'pincha KMP algoritmidan tezroq natija beradi, ayniqsa alohida strukturalangan matnlar uchun. Biroq Boyer-Moore'ni implementatsiya qilish murakkabroq bo'lib, unga ko'proq qo'shimcha yordamchi massivlar kerak bo'ladi. KMP esa nisbatan soddaroq tuzilgan va barqaror ishlashni ta'minlaydi, ayniqsa bir xil yoki takrorlanuvchi belgilar ko'p bo'lgan namunalar ustida.

Yana bir mashhur algoritm — Rabin-Karp algoritmidir. U hash funksiyalar yordamida matnning har bir substringini namunaga mos kelishini tekshiradi. Bu yondashuv o'rtacha holatda tez ishlasa ham, eng yomon holatda vaqt murakkabligi $O(n \cdot m)$ ga yetadi, chunki kolliziyalar (hash qiymatlarining mos kelishi) bo'lishi mumkin. KMP algoritmidagi esa deterministik yondashuv bo'lgani uchun bunday noaniqlik yo'q va har doim barqaror natija beradi.

Umuman olganda, KMP algoritmi barqaror ishlashi, aniq vaqt murakkabligi va nisbatan soddaligi bilan ajralib turadi. U yirik tizimlar, real vaqt ilovalari yoki resurslar cheklangan muhitlarda qo'llanilish uchun juda mosdir. Boyer-Moore murakkabroq

bo'lsa-da, katta matnlar va kam moslik hollari uchun qulay, Rabin-Karp esa hash funksiyalarga bog'liq bo'lgan holatlarda ishlatiladi. Shunday qilib, KMP algoritmi — bu soddalik va samaradorlikni birlashtirgan muvozanatli yondashuvdir.

Knut-Morris-Pratt (KMP) algoritmi matn ichida namunani izlash masalasida samarali va ishonchli yechim hisoblanadi. Uning asosiy afzalligi — ortiqcha taqqoslashlarni kamaytirish orqali qidiruv jarayonini sezilarli darajada tezlashtirishi va natijada murakkabligini $O(n + m)$ darajasida ushlab turishidir. Bu algoritm nafaqat nazariy jihatdan mustahkam asosga ega, balki amaliyotda ham keng qo'llaniladi. Matn tahlili, bioinformatika, xavfsizlik tizimlari va plagiat aniqlash kabi sohalarda KMP algoritmining samaradorligi va barqarorligi uning asosiy ustunliklari hisoblanadi. Shu bilan birga, KMP algoritmini boshqa qidiruv algoritmlari bilan solishtirganda uning soddaligi va aniq ishlash mexanizmi uni dasturchilar orasida juda mashhur qiladi.

Taklif qilish mumkinki, kelajakda KMP algoritmining imkoniyatlarini kengaytirish maqsadida, uni parallel hisoblash resurslari yordamida optimallashtirish yoki boshqa ilg'or string qidiruv usullari bilan kombinatsiyalash ustida tadqiqotlar olib borilsin. Shuningdek, yangi dasturlash tillari va platformalarda KMP algoritmining samarali kutubxonalarini yaratish ham foydali bo'ladi. Bu nafaqat algoritmni kengroq auditoriyaga yetkazishga, balki turli muhitlarda uning qo'llanilishini osonlashtirishga xizmat qiladi. Umuman olganda, KMP algoritmi hozirgi kunda ham dolzarb mavzu bo'lib, matn qidirish sohasida hal qiluvchi rol o'ynashda davom etmoqda. Shu sababli uning chuqur tahlili va amaliy qo'llanilishi bo'yicha ilmiy izlanishlar davom ettirilishi lozim.

Foydalanilgan adabiyotlar

1. Knuth, D. E., Morris, J. H., & Pratt, V. R. (1977). *Fast Pattern Matching in Strings*. SIAM Journal on Computing.
2. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms* (3rd ed.). MIT Press.
3. Gusfield, D. (1997). *Algorithms on Strings, Trees, and Sequences*. Cambridge University Press.

4. Aho, A. V., & Ullman, J. D. (1995). *Foundations of Computer Science*. W. H. Freeman.
5. Brassard, G., & Bratley, P. (1996). *Fundamentals of Algorithmics*. Prentice-Hall.
6. Manber, U. (1989). *Introduction to Algorithms: A Creative Approach*. Addison-Wesley.
7. Sedgewick, R., & Wayne, K. (2011). *Algorithms* (4th ed.). Addison-Wesley.
8. Levitin, A. (2012). *Introduction to the Design and Analysis of Algorithms*. Pearson.
9. Skiena, S. (2008). *The Algorithm Design Manual*. Springer.
10. Gokhale, S. S. (2017). *Data Structures and Algorithms*. University Science Press.
11. GeeksforGeeks. (n.d.). *KMP Algorithm for Pattern Searching*. <https://www.geeksforgeeks.org/kmp-algorithm-for-pattern-searching/>
12. Rosetta Code. (n.d.). *KMP string search*. https://rosettacode.org/wiki/KMP_string_search
13. Microsoft Docs. (n.d.). *String Searching Algorithms in C#*. <https://docs.microsoft.com>
14. Horowitz, E., Sahni, S., & Rajasekaran, S. (2007). *Fundamentals of Computer Algorithms*. University Press.
15. Baeza-Yates, R., & Ribeiro-Neto, B. (2011). *Modern Information Retrieval: The Concepts and Technology behind Search*. Addison-Wesley.