Київський національний університет імені Тараса Шевченка

Факультет комп`ютерних наук та кібернетики

Кафедра інтелектуальних інформаційних систем

Алгоритми та складність

Лабораторна робота 2

Виконав студент 2-го курсу

групи К-29

Прокопчук Роман

2021

**Завдання**

Розробіть алгоритм, який за лінійний час визначав би, чи є текстовий

рядок Т циклічним зсувом іншого рядка Т\* (наприклад, abc та cab).

**Теорія**

Доцільним до цієї задачі буде застосувати алгоритм Кнута-Морріса-Пратта, який в найгіршому випадку буде працювати за 2n.

Але для того, щоб перейти до його опису, потрібно розглянути поняття префікс-функції. Префікс-функцією рядка (S,i) називається довжина найбільшого префікса рядка S[1..i], який не збігається з цим рядком і одночасно є його суфіксом. Для рядка S зручно представляти префікс функції у вигляді масиву розміром S.length, поклавши (S, 1) = 0.

Наприклад, префікс-функція для шаблону “ababaca”: [0, 0, 1, 2, 3, 0, 1]:

“a”, “ab” не містять нетривіального префікса = суфіксу

“aba”: префікс довжини 1 співпадає з суфіксом

“abab” : префікс довжини 2 співпадає з суфіксом

“ababa” : префікс довжини 3 співпадає з суфіксом

“ababac” не містить нетрив. префікса = суфіксу

“ababaca” : префікс довжини 1 співпадає з суфіксом

Нехай S=T, L=T\* (T, T\* - рядки, які задано в умові)

Наш алгоритм буде використовувати префікс-функцію, але суфіксом буде рядок L, а префіксом – T. Обчисливши її, ми знайдемо певну «опорну точку», за допомогою якої визначимо результат(див.алгоритм)

**Алгоритм**

Нехай S=T, L=T\* (T, T\* - рядки, які задано в умові), [0..n-1] – префікс-функція, n-розмір рядка, i = 1 – індекс, len – довжина найдовшого префікса

* Поки i менше за n, перевіряємо, чи рівні T[i] та L[len].
* Якщо так, то в [0..n-1] кладемо значення len + 1, збільшуємо len та і на одиницю. Інакше, перевіряємо, чи len дорівнює дорівнює. Якщо так, то [i] = 0, збільшуємо і на одиницю, інакше len присваюємо значення [len - 1].

Таким чином, ми обчислили префікс-функцію. Далі вже йде перевірка на зсув

Нехай k = [n-1] – «опорна точка»

* Поки k менше за n, перевіряємо, чи L[k] дорівнює T[i]. Якщо так, то збільшуємо k та і на одиницю. Інакше, повертаємо false.

Якщо ми вийшли з циклу, не повернувши false, то повертаємо true.

**Складність**

Як зазначалося раніше, алгоритм виконує 2n операцій у найгіршому випадку(n – обчислення префікс-функції, n-перевірка на зсув), тобто, складність алгоритму .

**Мова програмування**

C++

**Модулі програми**

bool AreCyclicShift(const std::string& source, const std::string& shifted)

Функція, яка безпосередньо перевіряє, чи є рядок shifted циклічним зсувом рядка source.

**Інтерфейс користувача**

Вхідні дані вводяться і виводяться у консоль

**Тестові приклади**

|  |  |
| --- | --- |
| **input** | **output** |
| abc cab  abcd adcb  00010100 10100000  aaa aaa  banana anaban  100001 000101 | Yes  No  Yes  Yes  Yes  No |

**Висновки**

Застосування алгоритму Кнута — Морріса — Пратта є більш ефективним по складності підходом до розв’язання задачі, аніж інші, зокрема розв’язання з прямим перебором, складність якого є .

**Література**

* Лекція 8
* Donald Knuth; James H. Morris, Jr, Vaughan Pratt (1977). Fast pattern matching in strings. *SIAM Journal on Computing* **6** (2): 323–350.