Київський національний університет імені Тараса Шевченка

Факультет комп`ютерних наук та кібернетики

Кафедра інформаційних систем

Лабораторна робота №8

“Алгоритми пошуку зразка в текстовому рядку”

Виконала студентка 2го курсу

Групи К-29

Шарова Ірина Валеріївна

2018

**Завдання:**

Реалізуйте алгоритми пошуку зразка в текстовому рядку: наївний, Хорспула, Боєра-Мура, КПМ та Рабіна-Карпа і порівняйте їх ефективність. Виконайте пошук зразків різної довжини: випадкового бінарного зразка у випадковому бінарному тексті та випадкового слова у природному тексті на цій мові.

**Модулі програми:**

* NaiveSearchAlgorithm(string str, string subStr)

Наївний алгоритм пошуку.

Повертає індекс першого входження підрядка subStr в рядок str і -1 якщо subStr не міститься в str.

Вирівняємо шаблон з початком тексту, і посимвольно порівнюємо їх зліва направо. Якщо всі m символів рівні, то входження знайдено, інакше зсуваємо шаблон на один символ праворуч і посинаємо процедуру знову.

Середня оцінка складності буде О(n), в найгіршому випадку алгоритм деградує до O((n-m+1)m).

* HorspoolAlgorithm(string str, string subStr)

Алгоритм Хорспула

Повертає індекс першого входження підрядка subStr в рядок str і -1 якщо subStr не міститься в str.

Зразок рухається по тексту зліва направо, але порівняння відбувається справа наліво. В разі неспівпадіння це дозволяє зсунути зразок одразу на декілька символів(на скільки саму визначає допоміжна функція ShiftTable)

Час роботи в найгіршому випадку буде O(nm), в середньому час роботи O(n).

* BuildShiftTable(string str, int shift[])

Допоміжна функція, яка будує таблицю зсувів для алгоритмів Хорспула і Боєра-Мура.

Час роботи O(m)

Спочатку всі значення таблиці ініціалізуються довжиною зразка, потім зразок переглядається зліва направо до передостаннього символа і відповідний елемент таблиці записується відстанню цього символа до правого кінця зразка.

* BoyerMooreSearchAlgorithm(const string &where, const string &what)

Алгоритм Боєра-Мура.

Вважається найефктивншим з алгоритмів загального призначення.

Зазвичай час роботи сублінійний, в найгіршому випадку працює за O(nm).

Спочатку для заданого шаблона будується таблиця неспівпадаючих символів і таблиця не співпадаючих префіксів.

Зсув відбуваєьбся на кількість символів d, яке обчислюється наступним чином:

d = d1 = max{t(c) – k, 1}, k=0

max{d1, d2}, k>0

t(c) – значення зсуву для символа с в таблиці неспіпадаючих символів, k – кількість символів, що співпали, d2 – значення з таблиці не співпадаючих суфіксів.

* KMPAlgorihm (string s, string p)

Алгоритм Кнута-Морріса-Пратта.

Час роботи – лінійний.

Порівняння відбуваються зліва направо

Визначається префікс-функція рі така,що:

pi[q] = max{k: k < q && Pk є суфіксом Pq}, тобто pi[q] є довжиною гайбільшого префікса зразка, який є істинним суфіксом рядка Pq.

Зсув зразка відбувається згідно з префікс функцією.

* RabinKarpAlgorithm(string str, string subStr, vector<int>&answer)

Алгоритм Рабіна-Карпа

Час роботи в найкращому випадку складає О(n), в найгіршому - O(nm).

Алгоритм намагається пришвидшити пошук зразка, використовуючи хеш-функцію. Хеш-функція – функція, що спів ставляє кожній строці певне число, таким чином, що кожна строка маж унікальний номер.

Алгоритм порівнює хеші зразка і всіх підрядків довжини m у рядку.

**Тестові приклади:**

* Вхід

Sabcabc, abc

Вихід: 1

* Вхід

abcabc, c

Вихід: 2

**Література:**

* Лекція №8
* <https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%91%D0%BE%D0%B9%D0%B5%D1%80%D0%B0_%E2%80%94_%D0%9C%D1%83%D1%80%D0%B0>
* https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC\_%D0%A0%D0%B0%D0%B1%D0%B8%D0%BD%D0%B0\_%E2%80%94\_%D0%9A%D0%B0%D1%80%D0%BF%D0%B0