Київський національний університет імені Тараса Шевченка

Факультет комп’ютерних наук та кібернетики

Лабораторна робота №8

Виконав: Олексишин Олексій

Група К-28

Кафедра інтелектуальних програмних систем

**2019**

**Умова лабораторної роботи №8**

Реалізуйте алгоритми пошуку зразка в текстовому рядку: наївний, Хорспула, Боєра-Мура, КПМ та Рабіна-Карпа.

**Алгоритми**

**1) Наївний алгоритм.**

Наївний алгоритм проходиться по символах рядка, в якому ми шукаємо підрядок та шукає співпадіння першого символу, якщо він співпадає, то перевіряється другий і т.д.

**2) Алгоритм Хорслупа.**

Спочатку будується таблиця зміщень для шуканого шаблону. Поєднується початок тексту (рядки) і шаблона, перевірка починається з останнього символу шаблону.

Якщо останній символ шаблону і відповідний йому при накладенні символ рядка не збігаються, то зразок зрушується щодо рядка на величину, отриману з таблиці зміщень. Причому символ береться з рядка (а не з шаблону), відповідний зсув знаходиться в таблиці. Проводиться зрушення і знову починається перевірка з останнього символу.

Якщо ж символи збігаються, проводиться порівняння передостаннього символу шаблону і т.д. Якщо всі символи шаблону збіглися з накладеними символами рядки, значить, підрядок знайдена, і пошук закінчено. Якщо ж якийсь (не останній) символ шаблону не збігається з відповідним символом рядка, шаблон зсувається на один символ вправо, і перевірка знову починається з останнього символу. Весь алгоритм виконується до тих пір, поки або не буде знайдено входження шуканого зразка, або не буде досягнуто кінець рядка.

**3) Алгоритм Боєра-Мура**

Скануємо зліва направо, як і в наївниму алгоритмі порівнюється початок рядка та підрядка. Якщо всі символи співпадають, то місце підрядка знайдено і алгоритм закінчує свою дію. Якщо ж якийсь символ підрядка не співпадає з символо рядка, то підрядок зсовується на декілька символів направо так, щоб під стоп-символом (символ, на якому знайдена невідповідність) був символ такий символ у підрядку, якщо ж такого символу там немає, то зсовуємо на довжину підрядка.

Для того, щоб можна було швидко визначити на скільки потрібно зсовувати підрядок потрібно обчислити таблицю суфіксів (Для кожного можливого суфікса S з підрядкавказати найменшу величину, на яку потрібно зрушити вправо шаблон, щоб він знову збігся з *S*. Якщо такий зсув неможливий, ставиться довжина підрядка (в обох системах нумерації)).

**4) Алгоритм Кнута-Морріса-Пратта**

Будемо проходитися та шукати співпадіння так, як у наївному алгоритмі, тільки перевіряти будемо не кожен можлий підрядок, а багато з них будемо пропускати. Для цього будемо використовувати такий алгоритм: префікс рядка, якого ми шукаємо може співпадати з суфіксом підрядка, з яким ми порівнюємо, тому зсовувати будемо так, щоб ці префікс та суфікс співпали, що приведе до сильної оптимізації пошуку. Для цього перед виконанням порівняння обрахуємо префікс-функції для шуканого рядка (функція, яка обраховує масив, кожний елемент якого рівний довжині найдовшого префікса, який починається з позиції *i* суфікса цього рядка).

**5) Алгоритм Рабіна-Карпа**

Будемо порівнювати підрядки не по символах, а їх кеші. Обчислення хеша підрядка займає лінійний час, але після обчислення першого хеша можна рекурсивно обчислювати хеш наступного рядка, оскільки ми до вже обрахованого рядка забираємо перший сивол та додаємо останній. Тому Алгоритм буде працювати за лінійний час.

**Реалізація (мова С++)**

Програма складається з п’яти функцій **find(l1, l2),** кожна з яких приймає два рядки: заданий та шуканий та, використовуючи один з п’яти алгоритмів, описаних вище, перевіряє входження одного рядка в інший.

**Інтерфейс користувача**

Файл який містить код програми можна приєднати до власних програм та використовувати реалізовані функції з вказаним вище інтерфейсом.

**Літературні джерела**

* [https://ru.wikipedia.org/wiki/](https://ru.wikipedia.org/wiki/Z-функция)Z-функция
* [https://uk.wikipedia.org/wiki/](https://uk.wikipedia.org/wiki/Алгоритм_Кнута_—_Морріса_—_Пратта)Алгоритм\_Кнута\_—\_Морріса\_—\_Пратта
* <https://uk.wikipedia.org/wiki/Алгоритм_Бойера_Мура>
* [https://uk.wikipedia.org/wiki/](https://uk.wikipedia.org/wiki/Алгоритм_Бойєра_—_Мура_—_Хорспула)Алгоритм\_Бойєра\_—\_Мура\_—\_Хорспула
* <https://ru.wikipedia.org/wiki/Поиск_подстроки>