Лабораторная работа № 4 по курсу дискретного анализа: поиск подстроки в строке

Выполнил студент группы 08-207 МАИ *Хренов Геннадий.*

Условие

1. Необходимо реализовать один из стандартных алгоритмов поиска образцов для указанного алфавита.

2. Вариант алгоритма: Поиск одного образца при помощи алгоритма Бойера-Мура.

3. Вариант алфавита: Слова не более 16 знаков латинского алфавита (регистронезависимые)

Метод решения

Для решения реализации алгоритма Бойера-Мура я написал две вспомогательные функции, соответствующие правилам плохого символа и хорошего суффикса. Функция плохого символа составляет map, хранящий последнее вхождение для каждого символа алфавита, функция хорошего суффикса составляет вектор смещений, таких, что совпавший суффикс текста снова совпал бы с шаблоном, при этом идущий дальше символ шаблона теперь отличается от старого несовпавшего символа. Общий принцип алгоритма сводится к трём вещам: сравнению текста и образца происходит справа налево, и двум правилам, описанным выше. Смещение патерна определяется как наибольшее из смещений, предложенных правилами хорошего суффикса и плохого символа. Также для модификации алгоритма я использовал правило Галиля, которое заключается в том, что при нахождении вхождения необязательно по новой сравнивать элементы, которые уже прошли проверку. Для записи ответа я выделил вектор пар, который ставит соответствие между номером слова в тексте и его строкой, а также номером в строке.

Описание программы

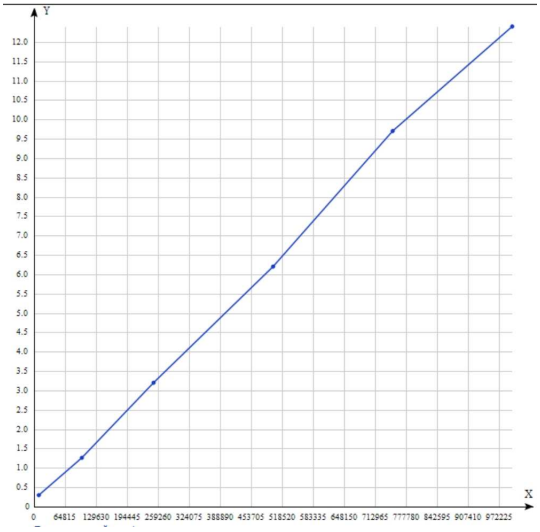
Программа состоит из файла lab4.cpp Основные функции:
Max,Min - нахождение максимального, минимального элемента из двух
BadChar - правило плохого символа
GoodSuffix - правило хорошего суффикса
Low - регистроонезависимость
Read - чтение патерна, текста и присваивание словам порядковых номеров

Дневник отладки

1-3 - Ошибка выполнения. Заменил return -1 на 0
4-5 - неправильный ответ. Пустые строки текста не считались как полноценные строки. Заменил алгоритм считывания данных.
6 - Превышено реальное время работы. Долгое чтение. Модифицировал алгоритм считывания.

Тест производительности

количество элеменнтов; время(с)
(10000; 0,296)
(100000; 1,26)
(250000; 3,2)
(500000;6,2)
(750000; 9,7)
(1000000; 12,4)



На графике просматривается линейная сложность.

Недочёты

Алгоритм можно модифицировать, добавив улучшенное правило плохого символа, которое будет хранить все вхождения букв(а не только последней) в образец, и сдвиг в данном случае будет совмещать несовпавщий символ и с ближайщим слева таким же символом. Правило хорошего суффикса реализовано на Z-функции, что требует дополнительных затрат памяти.

Выводы

Алгоритм Бойера-Мура имеет несколько различных методов реализаций и модификаций, что позволяет выбирать наиболее подходящую сортировку по затрате памяти и времени. Из плюсов - алгоритм не требует предварительную обработку текста. Из минусов - возможны большие затраты по памяти при большом алфавите, а также сложный и затратный препроцессинг.