Лабораторная работа № 4 по курсу дискретного анализа: поиск образцов

Выполнил студент группы М8о-207Б-18 МАИ *Токарев Никита*.

Условие

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

2. Вариант алгоритма: Поиск одного образца основанный на построении Z-блоков. Вариант алфавита: Слова не более 16 знаков латинского алфавита (регистронезависимые)

Метод решения

1. Считываю данные в формате P&T, где P - образец, T - текст, & - сентинел

2. Высчитываю значение Z-функции для элементов P&T (от второго до последнего элемента в данной строке). Алгоритм реализован с использованием r и l: правая и левая граница максимального Z-блока. При анализе элемента принадлежащего данному Z-блоку, берется информация, полученная при пошлых сравнениях.

3. Если значение Z-функции элемента принадлежащего T равно значению длины P, то вывожу строку и столбец, в которой находился данный элемент

Описание программы

В данной работе я создал класс TZBlocks, в котором параметрами являются: число равное размеру образца,число равное размеру текста, а также вектор умных указателей, которые содержат структуру TData. В структуре TData хранится слово, его локация(номер строки и номер элемента в строке), а также значение Z-функции для данного элемента, которое при создании объекта равно 0. В классе TZBlocks реализованы такие функции:

1. void Read(): В данной функции происходит заполнение вектора умных указателей необходимой структурой(TData) в нужной форме(P&T, где P - образец, T - текст, & - сентинел), подсчет размера образца и текста.

2. void BuildZBocks(): В данной функции вычисляется Z-функция для всех элементов вектора, кроме первого. В функции реализован нетривиальный алгоритм, в котором при анализе элемента принадлежащего Z-блоку, берется число, которое получено в результате прошлых сравнений. Таким образом, количество прямых срвнений уменьшается.

3. void SimpleSearch(): В данной функции проводится анализ значений Z-функции элементов принадлежащих T и вывод результата.

4. void Result(): В данной функции выполняется вызов функций: void BuildZBocks(), void SimpleSearch().

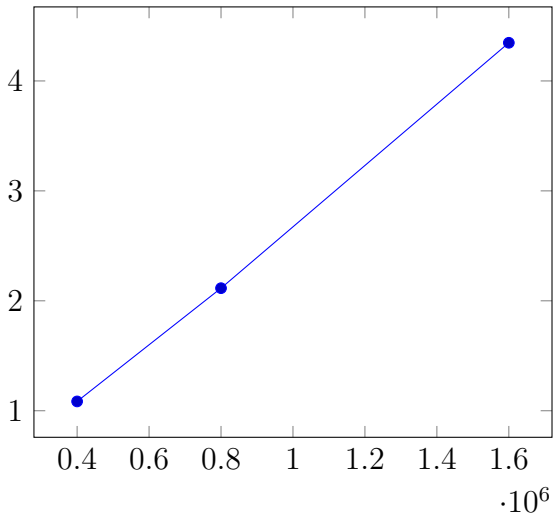
Дневник отладки

При тестировании были замечены такие ошибки: была ошибка компиляции. Затем же я просто модифицировал готовую работу.

1. Ошибки компиляции -> исправлял согласно стандартам C++;

Тест производительности

В данных тестах используется строки только из 50 элементов, сгенерированные рандомно. Слова генерируются рандомно,размер слова: от 0 до 16 букв.



Пояснения к графику:

1. За 1.0848c обрабатывается 400000 слов

2. За 2.11492c обрабатывается 800000 слов

3. За 4.34717c обрабатывется 1600000 слов

Выводы

В ходе данной работы я познакомился с таким алгоритмом: поиск подстроки в строки и с использованием Z-блоков. Также как мне известно с использованием этого алгоритма

работают и другие алгоритмы, например Ахо-Корасик. Также алгоритм построения Z-блоков делится на тривиальную реализацию и нетривиальную реализацию. Соответсвенно в нетривиальной реализации используются числа l и r, характеризующие позиции начала и конца максимального Z-блока. Данная нетривиальная реализация позволяет сравнивать элементы с сложностью O(n).