

Лабораторная работа № 4 по курсу дискретного анализа: Поиск образца в строке

Выполнил студент группы М8О-208Б-20 МАИ *Попов Матвей*.

Условие

1. **Упрощённый вариант:** поиск одного образца в тексте с помощью алгоритма Z-блоков.
2. **Алфавит:** строчные латинские буквы.

Метод решения

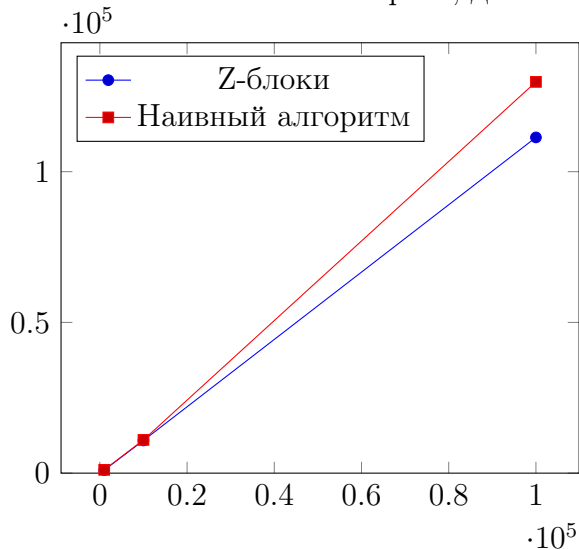
Создадим строку, состоящую из образца и текста, разделённых символом, не входящим в алфавит, и определим Z-функцию этой строки. Индекс элементов Z-функции, равных длине образца, будет определять позицию вхождения образца в тексте.

Дневник отладки

1. Программа была написана примерно за 20 минут без ошибок, это же не Ахо-Корасик с шакальным вводом.

Тест производительности

Сравнение производительности производилось с наивным алгоритмом поиска подстроки. Во всех графиках по оси Y отложено время выполнения (в миллисекундах), по оси X — количество символов в строке, делённое на 50.



Кол-во строк	Z-блоки	Наивный алгоритм
1000	1007	1097
10000	10840	11028
100000	111367	129790

Таким образом, алгоритм Z-блоков оказался немного быстрее наивного алгоритма, так как длина образца во всех тестах была сравнительно небольшой, однако стоит понимать, что с увеличением длины образца будет увеличиваться и разница во времени работы алгоритмов.

Выводы

Проделав лабораторную работу, я реализовал алгоритм поиска подстроки в строке, построение Z-функции работает за $O(n + m)$, где n — длина образца, а m — длина текста, а поиск всех вхождений работает за $O(m)$. Это гораздо эффективнее наивного алгоритма, который в худшем случае работает за $O(nm)$. Также положительной чертой этого алгоритма является то, что он относительно прост для понимания.