

Лабораторная работа №5 по курсу дискретного анализа: Суффиксные деревья

Выполнил студент группы М8О-312Б-22 МАИ *Корнев Максим*.

Условие

Вариант: 2

Найти в заранее известном тексте поступающие на вход образцы с использованием суффиксного массива.

Метод решения

Я решил задачу построения суффиксного дерева с помощью алгоритма Укконена, чтобы эффективно искать подстроки в строке. Сначала реализовал класс для узлов дерева, потом суффиксное дерево, где шаг за шагом добавлял символы строки. Далее на основе дерева построил суффиксный массив и применил бинарный поиск для нахождения всех вхождений подстроки.

Описание программы

Для реализации алгоритма были реализованы следующие функции и структуры:

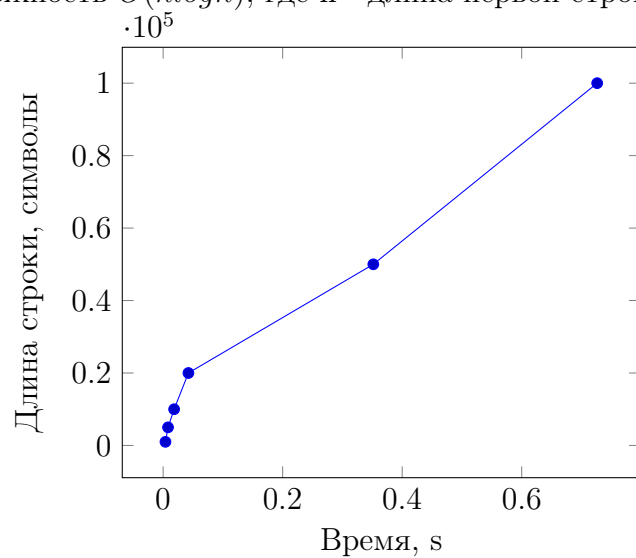
- **class Node:** представляет узел суффиксного дерева. Хранит ссылки на дочерние узлы, указатели на начало и конец строки, а также ссылку на суффиксную связь
- **class SufTree:** Реализует суффиксное дерево с помощью алгоритма Укконена. Постепенно расширяет дерево, добавляя символы строки, и поддерживает активную точку (узел, длину и активное ребро). Содержит методы для построения дерева, управления суффиксными ссылками и поиска.
- **class SufArr:** построен на основе суффиксного дерева. Хранит суффиксный массив и предоставляет метод `Search` для поиска всех вхождений подстроки с использованием бинарного поиска `equal_range`.

Дневник отладки

1. Был получен TL на тесте №1. Немного напутал в логике функции суффиксного дерева.
2. Был получен WA на тесте №1. При поиске в суффиксном массиве когда добавлял левую границу случайно добавлял 1, а не l.

Тест производительности

Алгоритм построения суффиксного массива работает за "суперлинейное" время, то есть сложность $O(n \log n)$, где n - длина первой строки.



Выводы

В ходе выполнения пятой лабораторной работы по курсу «Дискретный анализ» я изучил различные подходы к поиску паттернов в тексте, такие как использование суффиксного дерева и суффиксного массива. Суффиксное дерево оказывается особенно полезным, когда текст остаётся неизменным, а для поиска регулярно поступают новые паттерны. Однако этот метод требует много памяти, что привело к разработке суффиксного массива. Суффиксный массив выполняет те же задачи, но использует $O(n)$ памяти, где n — длина текста.