Лабораторная работа №5 по курсу дискретного анализа: Суффиксные деревья

Выполнил студент группы М8О-312Б-22 МАИ *Юрков Евгений*.

Условие

Вариант: 5

Найти самую длинную общую подстроку двух строк с использованием суффиксного дерева.

Формат ввода: Две строки.

Формат вывода: На первой строке нужно распечатать длину максимальной общей подстроки, затем перечислить все возможные варианты общих подстрок этой длины в порядке лексикографического возрастания без повторов.

Метод решения

Для построения суффиксного дерева за линейное время был использован алгорим Укконена. Чтобы найти самую длинную общую подстроку необходимо было построить обобщенное суффиксное дерево, для этого в каждой вершине хранятся номера строк, которые содержат данный суффикс. Поиск осуществляется по следующему принципу: обходом в глубину находятся все подстроки вершин, которые принадлежат сразу всем строкам и выбирается самая длинная. Данный алгоритм может также использоваться для поиска максимальной общей подстроки более чем двух строк.

Описание программы

Все алгоритмы реализованы в классе `SuffTree`, содержащем следующие методы:

- `SuffTree` - конструктор, строящий дерево из одной или нескольких строк;
- `add` - метод дополняющий обобщенное суффиксное дерево ещё одной строкой;
- `max_com_substr` - поиск самой длинной общей подстроки для всех строк, содержащихся в дереве;
- `Print` - вывод структуры дерева, для удобной отладки;

Дневник отладки

1. исправлено создание суффиксных ссылок, чтобы они не указывали на корень
2. исправлено разделение вершины, чтобы вершины, имеющие ссылку на неё, не теряли её
3. добавлена проверка и исправление массива, указывающего на то, к какой строке относится вершина, так как из за перехода по ссылкам не все вершины отмечали свои строки.

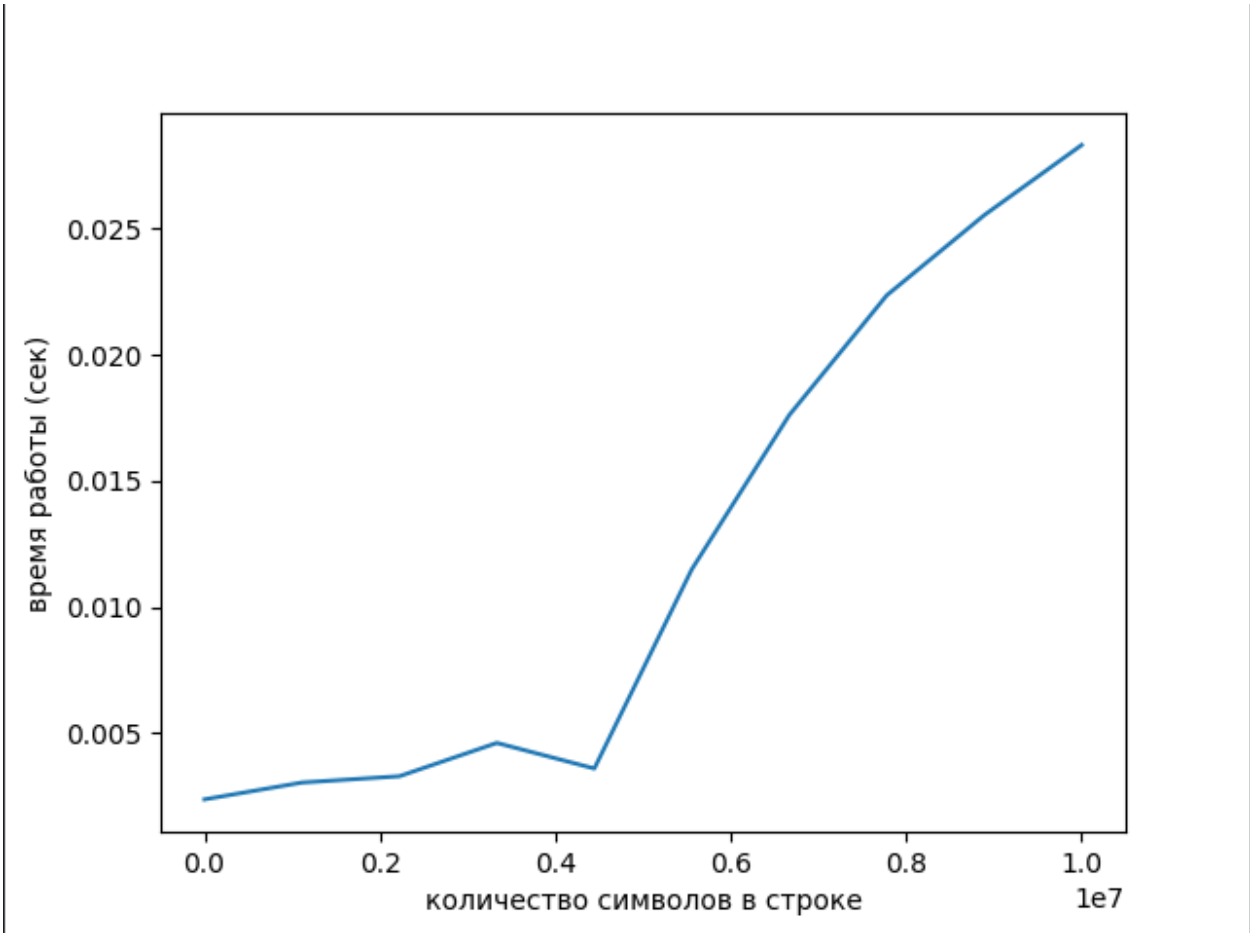

Рис. 1: График зависимости времени работы программы от длины строки

Тест производительности

Алгоритм Укконена для построения суффиксного дерева работает за линейное время: $O(n)$, где n – длина строки.

Выводы

В ходе данной лабораторной работы была реализована программа, предназначенная для поиска самой длинной общей подстроки двух строк с помощью суффиксного дерева. Алгоритм Укконена для построения суффиксного дерева работает за линейное время $O(n)$, в отличие от наивного, который работает за $O(n^3)$. Для поиска максимальной общей подстроки было применено обобщенное суффиксное дерево для двух строк, что позволило эффективно решать задачу.

Суффиксные деревья очень полезны в анализе текстов и поисковых задачах. Они применяются для эффективного поиска подстрок, сжатия данных, а также для построения суффиксных массивов. Их важность заключается в способности быстро находить различные паттерны в строках, обеспечивая высокую производительность даже при больших объемах данных.

Несмотря на свою скорость, суффиксные деревья занимают большое количество памяти, поэтому в случаях, где память важнее используют суффиксные массивы. Суффиксные массивы работают дольше, но занимают гораздо меньше памяти.