Лабораторная работа № 5 по курсу дискретного анализа: суффиксные деревья

Выполнил студент группы 08-307 МАИ *Боев Савелий*.

Условие:

Найти в заранее известном тексте поступающие на вход образцы, используя суффиксные деревья.

Метод решения

Функция **Insert** реализует добавление суффиксов в суффиксное дерево. При добавлении суффикса дерево обновляется таким образом, чтобы представлять все суффиксы исходного текста. Суффиксное дерево позволяет эффективно находить все вхождения образца в тексте за линейное время.

В функции **main** сначала считывается строка **text**, в которой будет выполняться поиск, а затем несколько строк **pattern**, каждую из которых мы хотим найти в тексте.

После этого вызывается функция для создания суффиксного дерева на основе **text**. Во время этого процесса каждый суффикс исходного текста добавляется в дерево.

Как только дерево готово, для каждого образца **pattern** вызывается функция **Find**. Эта функция начинает поиск с корня дерева и продвигается вглубь, следуя рёбрам, соответствующим символам образца. Если образец полностью найден в дереве, функция использует глубокий поиск для определения всех позиций в тексте, на которых начинается этот образец.

Затем, для каждой найденной позиции, программа выводит номер образца и соответствующие позиции в тексте, где этот образец начинается.

Описание программы

Инициализация суффиксного дерева: Программа начинается с создания структур данных для представления узлов и рёбер суффиксного дерева. Для

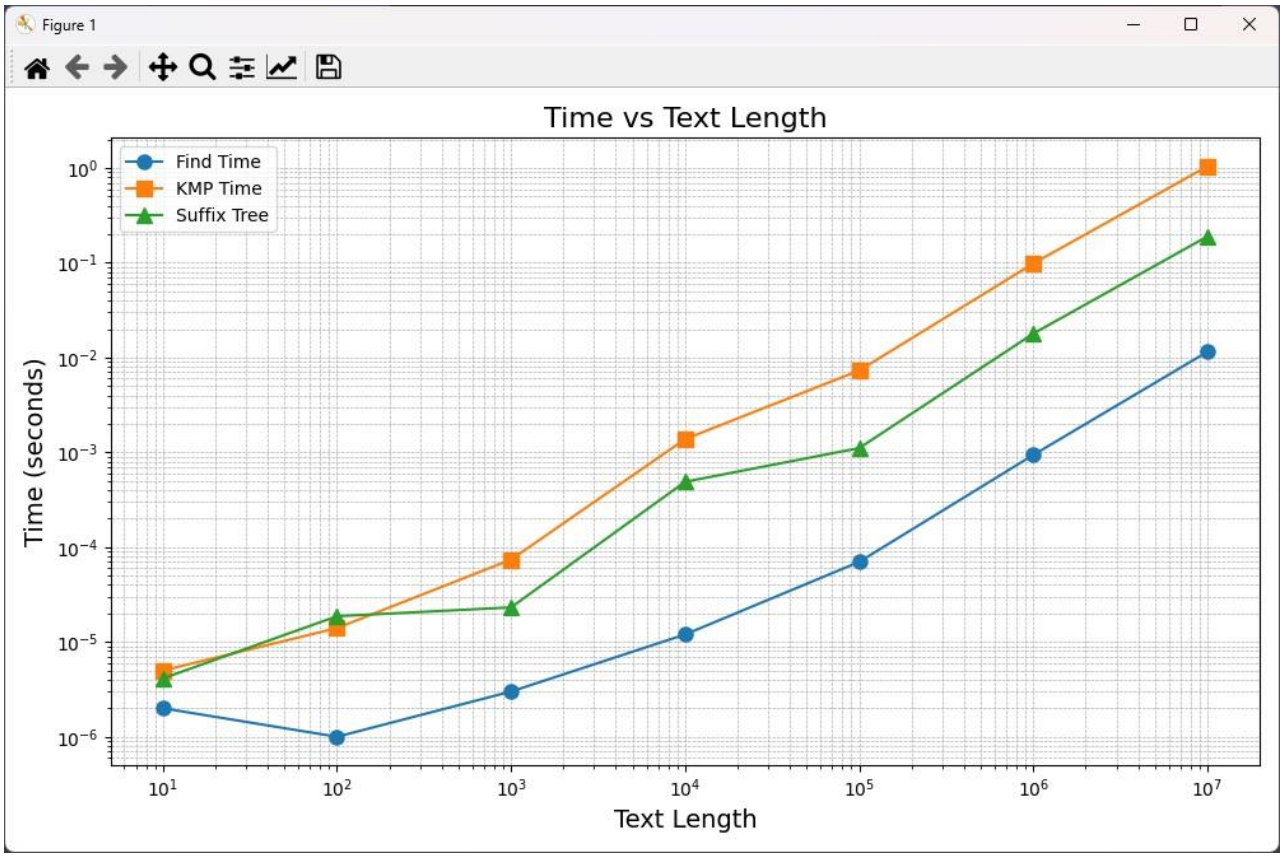
каждого ребра хранятся начальный и конечный индексы соответствующей подстроки исходного текста.

Построение суффиксного дерева: с помощью последовательного добавления всех суффиксов текста строится суффиксное дерево. При этом используется метод "активного узла" для оптимизации процесса вставки.

Поиск вхождений: при получении образца для поиска программа начинает искать его в суффиксном дереве, начиная от корня. Если образец полностью найден в дереве, программа использует глубокий поиск для определения всех позиций в тексте, на которых начинается образец.

Вывод результатов: программа выводит позиции в тексте, на которых начинаются найденные образцы.

Тест производительности



Производительность "Find": Встроенная функция поиска строки "Find" показывает отличную производительность на всех размерах входных данных. Она консистентно выполняется быстрее, чем остальные алгоритмы, особенно на больших размерах текста.

Производительность Кнута-Морриса-Пратта (KMP): Алгоритм Кнута-Морриса-Пратта становится значительно медленнее по сравнению с "Find" и суффиксными деревьями при увеличении размера текста. На самых больших размерах данных (10^7) KMP выполняется почти в 100 раз медленнее, чем встроенная функция "Find".

Производительность суффиксных деревьев: Алгоритм суффиксных деревьев показывает промежуточную производительность между "Find" и KMP. На больших размерах данных (10^7) суффиксное дерево выполняется в 17 раз быстрее, чем KMP, но медленнее, чем "Find".

Выводы

В данной лабораторной работе я реализовал поиск всех вхождений образца в

тексте с использованием суффиксного дерева. Суффиксное дерево является эффективным инструментом для решения различных задач на строках, включая поиск подстроки в тексте.