Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1
по курсу «Анализ алгоритмов»

«Расстояние Левенштейна и Дамерау-Левенштейна»

Студент Маслова Марина Дмитриевна

Группа ИУ7-53Б

Оценка (баллы) ____________

Преподаватель Волкова Лилия Леонидовна

2021 г.

Оглавление

Введение

Расстояние Левенштейна — минимальное количество операций вставки, удаления и замены символа, необходимых для превращения одной строки в другую. Если к указанным операциям добавить перестановку двух соседних символов, получим определение *расстояния Дамерау-Левенштейна* [1]. Поиск каждой из этих характеристик основан на рекуррентных вычислениях, то есть на вычислениях, которые используют уже вычисленные значения для вычисления новых.

Таким образом, задача поиска данных расстояний основывается на методе динамического программирования — разбиении задач на более мелкие и простые подзадачи такого же вида, решение которых проводится один раз и далее используется при решении других задач и подзадач [2]. Поэтому изучение, разработка и реализация алгоритмов поиска расстояний Левенштейна и Дамерау-Левенштейна позволит получить навыки использования данного метода.

Целью данной работы является изучение метода динамического программирования на материале алгоритмов Левенштейна и Дамерау-Левенштейна.

Для достижения поставленной цели необходимо выполнить следующие **задачи**:

- изучить алгоритмы Левенштейна и Дамерау-Левенштейна нахождения расстояния между строками;
- разработать алгоритмы поиска расстояния между строками;
- реализовать указанные алгоритмы;
- провести тестирование реализованных алгоритмов;
- провести сравнительный анализ алгоритмов по затрачиваемой памяти и процессорному времени работы реализации.

1 Аналитическая часть

В данном разделе представлено теоретическое описание алгоритмов поиска расстояния Левенштейна и Дамерау-Левенштейна, а также рассмотрены области их применения.

1.1 Расстояние Левенштейна

Расстояние Левенштейна[1] между двумя строками — это минимальное количество операций вставки, удаления и замены символа, необходимых для превращения одной строки в другую.

Цена операции может зависеть от её вида и/или от участвующих в ней символов, что отражает разную вероятность различных ошибок при вводе текста и т. п. Для решения задачи поиска расстояния между двумя страками необходимо найти последовательность применяющихся операций, такую, что суммарная их цена будет минимальной. При вычислении расстояния Левенштейна используются следующие цены:

– $w(a, a) = 0$ — цена совпадения двух символов;
– $w(a, b) = 1,\ a \neq b$ — цена замены символа a на символ b;
– $w(\lambda, a) = 1$ — цена вставки символа a;
– $w(a, \lambda) = 1$ — цена удаления символа a.

1.1.1 Рекурсивный алгоритм

В рекурсивном алгоритме поиска расстояния Левенштейна между двумя строками искомая величина вычисляется через соответсвующие величины подстрок, а рекуррентная формула выводится из следующих рассуждений:

1) для перевода пустой строки в пустую требуется ноль операций;

2) для перевода пустой строки в строку s требуется $|s|$ операций вставки (здесь и далее $|s|$ обозначает длину строки);

3) для перевода строки s в пустую строку требуется $|s|$ операций удаления;

4) для перевода строки s_1 в строку s_2 требуется выполнить некоторую последовательность операций удаления, вставки или замены, при этом операции в оптимальной последовательности можно произвольно менять места-

ми, так как две последовательные операции любых видов можно переставить, что доказывается простым перебором вариантов возможных пар, поэтому без ограничения общности можно считать, что операция над последним символом была произведена последней и цена преобразования строки s_1 в строку s_2 будет являться минимальной ценой из цен, полученных одним из следующих способов (пусть при этом s_1' и s_2' — строки s_1 и s_2 без последнего символа, соответственно):

– сумма цены преобразования строки s_1 в s_2' и цены проведения операции вставки, которая необходима для преобразования s_2' в s_2;

– сумма цены преобразования строки s_1' в s_2 и цены проведения операции удаления, которая необходима для преобразования s_1 в s_1';

– сумма цены преобразования из s_1' в s_2' и операции замены, предполагая, что s_1 и s_2 оканчиваются на разные символы;

– цена преобразования из s_1' в s_2', предполагая, что s_1 и s_2 оканчиваются на один и тот же символ.

Таким образом, для расчета расстояния Левенштейна между двумя строками s_1 и s_2 используется рекуррентная формула для расчета через подстроки:

$$D(s_1[1..i], s_2[1..j]) = \begin{cases} 0, & i = j = 0 \\ j, & i = 0,\ j > 0 \\ i, & j = 0,\ i > 0 \\ \min\{ & \\ \quad D(s_1[1..i], s_2[1..j-1]) + 1, & \\ \quad D(s_1[1..i-1], s_2[1..j]) + 1, & \\ \quad D(s_1[1..i-1], s_2[1..j-1]) + l(s_1[i], s_2[j]) & \\ \} & \end{cases}, \tag{1.1}$$

где величина $l(a, b)$ выражется формулой:

$$l(a, b) = \begin{cases} 0, & \text{если } a = b \\ 1, & \text{иначе} \end{cases}. \tag{1.2}$$

1.1.2 Матричный алгоритм

1.1.3 Рекурсивный алгоритм с кэшем

1.2 Расстояние Дамерау-Левенштейна

1.2.1 Рекурсивный алгоритм

1.3 Области применения алгоритмов

Литература

[1] Черненький В. М., Гапанюк Ю. Е. Методика идентификации пассажира по установочным данным // Инженерный журнал: наука и инновации. 2012. № 3. С. 30–39.

[2] Окулов С. М., Пестов О. А. Динамическое программирование. М.: БИНОМ. Лаборатория знаний, 2012. с. 296.