|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **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»**

|  |  |
| --- | --- |
|  |  |

Тема: \_\_Расстояние Левенштейна и Дамерау-Левенштейна\_\_\_\_\_

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ7И-52Б |  |  | Ву Хай Данг |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
| Преподаватель |  |  |  | Волкова Л.Л. |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |
| Преподаватель |  |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Строганов Ю.В. |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2023

Оглавление

Введение**1**

Type chapter title (level 2)2

Type chapter title (level 3)3

**Type chapter title (level 1)4**

Type chapter title (level 2)5

Type chapter title (level 3)6

# Введение

**Расстояние Левенштейна** - минимальное количество операций вставки одного символа, удаления одного символа и замены одного символа на другой, необходимых для превращения одной строки в другую.

Расстояние Левенштейна применяется в теории информации и компьютерной лингвистике для:

* исправления ошибок в слове
* сравнения текстовых файлов утилитой diff
* в биоинформатике для сравнения генов, хромосом и белков

**Расстояние Дамерау — Левенштейна** — это мера разницы двух строк символов, определяемая как минимальное количество операций вставки, удаления, замены и транспозиции (перестановки двух соседних символов), необходимых для перевода одной строки в другую. Является модификацией [расстояния Левенштейна](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D1%81%D1%81%D1%82%D0%BE%D1%8F%D0%BD%D0%B8%D0%B5_%D0%9B%D0%B5%D0%B2%D0%B5%D0%BD%D1%88%D1%82%D0%B5%D0%B9%D0%BD%D0%B0): к операциям вставки, удаления и замены символов, определённых в расстоянии Левенштейна добавлена операция [транспозиции](https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%B0%D0%BD%D1%81%D0%BF%D0%BE%D0%B7%D0%B8%D1%86%D0%B8%D1%8F_(%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)) (перестановки) символов.

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

Задачами данной лабораторной являются:

1. изучение алгоритмов Левенштейна и Дамерау - Левенштейна нахождения расстояния между строками;
2. создать ПО, реализующее перечисленные выше алгоритмы
3. провести сравнительный анализ реализаций алгоритмов по затраченному процессорному времени и памяти.
4. описание и обоснование полученных результатов в отчете о выполненной лабораторной работе, вы полненного как расчётно-пояснительная записка к работе.

# Аналитическая часть

* 1. Расстояние Левенштейна

Задача по нахождению расстояния Левенштейна заключается в поиске минимального количества операций вставки/ удаления/ замены для превращения одной строки в другую.

Введем следующие обозначения операций с указанием соответсвующих штрафов 𝑤:

— D (англ. delete) — удаление (𝑤(𝑎, 𝜆) = 1);

— I (англ. insert) — вставка (𝑤(𝜆, 𝑏) = 1);

— R (англ. replace) — замена (𝑤(𝑎, 𝑏) = 1, 𝑎 ̸= 𝑏);

Пусть 𝑆1 и 𝑆2 — две строки (длиной M и N соответственно) над некоторым алфавитом, тогда расстояние Левенштейна можно подсчитать по рекуррентной формуле 1.1.

D(i, j) = (1.1)

где m определяется следующим образом:

m(a, b) = (1.2)

С увеличением i, j , прямая рекурсивная реализация формулы (1.1) принимает вид

неэффективно с точки зрения времени выполнения, поскольку набор промежуточных значений 𝐷(𝑖, 𝑗) вычисляется много раз. Для решения этой проблемы

проблема, вы можете использовать матрицу для хранения соответствующих

промежуточные значения.

Чтобы избежать повторных вычислений,

будет использоваться матрица размера (𝑙𝑒𝑛𝑔𝑡ℎ(𝑆1) + 1)x((𝑙𝑒𝑛𝑔𝑡ℎ(𝑆2) + 1), где (𝑆) — длина

строка S. Значение в ячейке [𝑖, 𝑗] равно значению 𝐷(𝑆1[1...𝑖], 𝑆2[1...𝑗])

* 1. Расстояние Дамерау-Левенштейна

Расстояние Дамерау — Левенштейна измеряет минимальное количество операций (вставок, удалений, замен и перестановок символов), необходимых для преобразования одной строки в другую. Эта метрика часто используется в информационной теории, лингвистике и компьютерной науке для сравнения и анализа текстовых данных.

Сравнивая две метрики, расстояние Дамерау — Левенштейна учитывает возможность перестановки символов, что может быть полезным, например, при исправлении опечаток, где буквы могут быть перепутаны в словах. Это позволяет более точно измерить сходство между двумя строками и более эффективно обрабатывать различные виды текстовых данных.

Тогда для данного алгоритма определены следующие редакционные операции:

• вставка символа в произвольной позиции (I - Insert);

• удаление символа в произвольной позиции (D - Delete);

• замена символа на другой (R - Replace);

• транспозиция двух символов (M - Match).

Расстояние Дамерау-Левенштейна может быть найдено по формуле:

D(i, j) = (1.3)

* 1. Рекусивный алгоритм нахождения расстояния Дамерау-Левенштейна

Данный алгоритм использует для решения формулу 1.2 и является рекурсивным, а значит, для хранения промежуточных результатов используется стек. Кроме того, при этом подходе возникает проблема повторных вычислений, так как функция D(s1[1..i], s2[1..j]) будет выполняться несколько раз в разных ветвях дерева.

* 1. Рекусивный алгоритм нахождения расстояния Дамерау-Левенштейна с кэшем

Рекурсивный алгоритм заполнения можно оптимизировать по времени выполнения с использованием кеша. В качестве кеша используется матрица. Суть данной оптимизации заключается в параллельном заполнении матрицы при выполнении рекурсии. Изначально матрица инициализируется значениями, которые заведомо не могут быть получены в результате вычислений, например, -1

Вывод

В данном разделе были рассмотрены алгоритмы поиска расстояния Левенштейна и расстояния Дамерау-Левенштейна. В частности были приведены рекурентные формулы работы алгоритмов, объяснена разница между расстоянием Левенштейна и расстоянием Дамерау-Левенштейна.

1. Конструкторская часть

В этом разделе будут приведены требования к вводу и программе, а также схемы алгоритмов нахождения расстояний Левенштейна и Дамерау Левенштейна.

2.1 Разработка алгоритма поиска расстояния Левенштейна