

МГТУ им. БАУМАНА

ЛАБОРАТОРНАЯ РАБОТА №1

*Москва, 2020*

# Оглавление

<b>Введение</b>	<b>2</b>
<b>1 Аналитическая часть</b>	<b>3</b>
1.1 Некоторые теоретические сведения . . . . .	3
1.2 Расстояние Левенштейна . . . . .	3
1.3 Расстояние Дамерау-Левенштейна . . . . .	4
1.4 Вывод . . . . .	4
<b>2 Конструкторская часть</b>	<b>5</b>
<b>3 Технологическая часть</b>	<b>6</b>
3.1 Выбор яп, модули, тесты... . . . .	6
<b>4 Исследовательская часть</b>	<b>7</b>
<b>Заключение</b>	<b>8</b>

# Введение

В данной лабораторной работе мы познакомимся с расстоянием Левенштейна. Данное расстояние показывает нам минимальное количество редакторских операций (вставки, замены и удаления), которые необходимы нам для перевода одной строки в другую. Это расстояние помогает определить схожесть двух строк.

Какие ошибки человек может допускать при написании какого-то текста? Например, его пальцы могут нажимать на нужные клавиши не в том порядке. С этой проблемой поможет нам справиться расстояние Дамерау-Левенштейна. Данное расстояние задействует еще одну редакторскую операцию - транспозицию.

Практическое применение расстояние Левенштейна:

- Сравнение введенной строки со словарными словами в поисковой системе, такой как 'yandex' или 'google'
- Помогает найти разницу двух ДНК. Оценка мутации.

Целью данной работы является разбор и реализация алгоритма Дамерау-Левенштейна и Левенштейна.

В рамках выполнения работы необходимо решить следующие задачи:

1. Изучить алгоритмы Дамерау-Левенштейна и Левенштейна.
2. Реализовать изученные алгоритмы, а также матричную и рекурсивную реализацию алгоритма.
3. Подсчет времени поиска расстояния.
4. Сравнить временные характеристики, а также затраченную память.
5. Описать выбранную среду разработки и ЯП.

# 1 | Аналитическая часть

## 1.1 Некоторые теоретические сведения

При преобразовании одного слова в другое мы можем использовать следующие операции:

1. D (*от англ. delete*) - удаление.
2. I (*от англ. insert*) - вставка.
3. R (*от англ. replace*) - замена.

Будем считать стоимость каждой вышеизложенной операции - 1.

Введем понятие совпадения - M (*от англ. match*). Его стоимость будет равна 0.

## 1.2 Расстояние Левенштейна

Имеем две строки  $S_1$  и  $S_2$ , длиной M и N соответственно. Расстояние Левенштейна рассчитывается по приведенной ниже рекуррентной формуле.

$$D(S_1[1...i], S_2[1...j]) = \begin{cases} j, \text{ если } i == 0 \\ i, \text{ если } j == 0 \\ \min( \\ D(S_1[1...i], S_2[1...j-1]) + 1, \\ D(S_1[1...i-1], S_2[1...j]) + 1, \quad j > 0, i > 0 \\ D(S_1[1...i-1], S_2[1...j-1]) + \\ \quad \begin{cases} 0, \text{ если } S_1[i] == S_2[j] \\ 1, \text{ иначе} \end{cases} \\ ), \end{cases}$$

### 1.3 Расстояние Дамерау-Левенштейна

Как было написано выше, в расстоянии Дамерау-Левенштейна задействует еще одну редакторскую операцию - транспозицию Т (*от англ. transposition*). Рекуррентная формула Дамерау-Левенштейна представлена ниже.

$$D(S_1[1...i], S_2[1...j]) = \begin{cases} j, \text{ если } i == 0 \\ i, \text{ если } j == 0 \\ \min( \\ D(S_1[1...i], S_2[1...j-1]) + 1, \\ D(S_1[1...i-1], S_2[1...j]) + 1, \\ D(S_1[1...i-1], S_2[1...j-1]) + \\ \left[ \begin{array}{l} 0, \text{ если } S_1[i] == S_2[j] \\ 1, \text{ иначе} \end{array} \right. \\ D(S_1[1...i-2], S_2[1...j-2]) + 1, \text{ } i, j > 1, a_i = b_{j-1}, b_j = a_{i-1} \\ \left. \right), \end{cases} \quad j > 0, i > 0$$

### 1.4 Вывод

Мы познакомились с основополагающими материалами, которые в дальнейшем помогут нам при реализации алгоритмов Левенштейна и Дамерау-Левенштейна.

## 2 | Конструкторская часть

Требования ...

## 3 | Технологическая часть

### 3.1 Выбор яп, модули, тесты...

## 4 | Исследовательская часть



# Заключение

В этой лабораторной работе мы...