**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

отчет

**по лабораторной работе №3 (Вар. 6а)**

**по дисциплине «Построение и анализ алгоритмов»**

Тема: **Динамическое программирование**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 3388 |  | Глебова В.С. |
| Преподаватель |  | Жангиров Т.Р. |

Санкт-Петербург

2025

## Задание

Добавляется 4-я операция со своей стоимостью: транспозиция с заменой: два

последовательных символа меняются местами, а затем один из них может быть

заменён.

## Выполнение работы

Данный код реализует расширенный алгоритм Левенштейна для вычисления расстояния между двумя строками с учетом операций удаления, вставки, замены и транспозиции (перестановки двух соседних символов).

## Основные компоненты алгоритма

### Параметры функции

Функция extended\_levenshtein принимает:

* s, t - сравниваемые строки
* del\_cost - стоимость операции удаления (по умолчанию 1)
* ins\_cost - стоимость операции вставки (по умолчанию 1)
* repl\_cost - стоимость операции замены (по умолчанию 1)
* trans\_repl\_cost - стоимость операции транспозиции (по умолчанию 1)

### Инициализация

1. Создается матрица dp размером (m+1) x (n+1), где m и n - длины строк s и t соответственно.
2. Первая строка инициализируется стоимостью вставки соответствующих символов.
3. Первый столбец инициализируется стоимостью удаления соответствующих символов.

### Основной алгоритм

Для каждой пары символов из s и t:

1. Если символы совпадают, берется значение из диагональной ячейки без изменения.
2. Если символы различаются, вычисляются стоимости трех операций:
   * Удаление (значение из верхней ячейки + del\_cost)
   * Вставка (значение из левой ячейки + ins\_cost)
   * Замена (значение из диагональной ячейки + repl\_cost)
3. Выбирается минимальная из этих стоимостей.
4. Проверяется возможность транспозиции (если текущий и предыдущий символы в s соответствуют предыдущему и текущему символам в t):
   * Если символы просто переставлены, вычисляется стоимость транспозиции
   * Если символы отличаются, вычисляется стоимость транспозиции с заменой

### Отладочный вывод

При DEBUG = True выводится подробная информация о каждом шаге алгоритма:

* Инициализация матрицы
* Совпадения символов
* Варианты операций и их стоимости
* Транспозиции и их обработка
* Промежуточное состояние матрицы после обработки каждой строки
* Итоговая матрица расстояний

## Пример работы

Для входных данных "abc", "acb":

1. Инициализируется матрица 4x4
2. Обрабатывается первый символ 'a' - совпадение
3. Обрабатывается пара 'b' и 'c' - возможна транспозиция
4. Алгоритм обнаруживает, что "bc" можно преобразовать в "cb" транспозицией

## Преимущества реализации

1. Поддержка настраиваемых стоимостей операций
2. Подробный отладочный вывод для понимания работы алгоритма
3. Обработка транспозиций, что делает алгоритм более точным для реальных опечаток

## Ограничения

1. Использование дополнительной памяти O(m\*n) для хранения матрицы
2. Временная сложность O(m\*n) из-за двойного цикла
3. Транспозиция обрабатывается только для соседних символов