

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №3
по дисциплине «Построение и анализ алгоритмов»
Тема: Динамическое программирование. Редакционное расстояние
между строками.

Студент гр. 3388

Павлов А.Р.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2025

Цель работы:

Изучить принципы динамического программирования, использовать их для решения задач на поиск редакционного расстояния разного вида между строками.

Задание 1:

Над строкой ϵ (будем считать строкой непрерывную последовательность из латинских букв) заданы следующие операции:

- $\text{replace}(\epsilon, a, b)$ – заменить символ a на символ b .
- $\text{insert}(\epsilon, a)$ – вставить в строку символ a (на любую позицию).
- $\text{delete}(\epsilon, b)$ – удалить из строки символ b .

Каждая операция может иметь некоторую цену выполнения (положительное число). Даны две строки A и B , а также три числа, отвечающие за цену каждой операции. Определите минимальную стоимость операций, которые необходимы для превращения строки A в строку B .

Входные данные: первая строка – три числа: цена операции replace , цена операции insert , цена операции delete ; вторая строка – A ; третья строка – B .

Выходные данные: одно число – минимальная стоимость операций.

Sample Input:

```
1 1 1
entrance
reenterable
```

Sample Output:

```
5
```

Задание 2:

Над строкой ϵ (будем считать строкой непрерывную последовательность из латинских букв) заданы следующие операции:

- $\text{replace}(\epsilon, a, b)$ – заменить символ a на символ b .

- insert(ϵ , a) – вставить в строку символ a (на любую позицию).
- delete(ϵ , b) – удалить из строки символ b.

Каждая операция может иметь некоторую цену выполнения (положительное число). Даны две строки A и B, а также три числа, отвечающие за цену каждой операции.

Определите последовательность операций (редакционное предписание) с минимальной стоимостью, которые необходимы для превращения строки A в строку B.

Sample Input:

Первая строка – три числа: цена операции replace, цена операции insert, цена операции delete; вторая строка – A; третья строка – B.

Sample Output:

Первая строка – последовательность операций (M – совпадение, ничего делать не надо; R – заменить символ на другой; I – вставить символ на текущую позицию; D – удалить символ из строки); вторая строка – исходная строка A; третья строка – исходная строка B.

Задание 3:

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

Разработайте программу, осуществляющую поиск расстояния Левенштейна между двумя строками.

Пример:

Для строк pedestal и stien расстояние Левенштейна равно 7:

Сначала нужно совершить четыре операции удаления символа: pedestal -> stal.

Затем необходимо заменить два последних символа: stal -> stie.

Потом нужно добавить символ в конец строки: stie -> stien.

Реализация

Используется динамическое программирование на матрице: $dp[i][j]$ хранит минимальную стоимость преобразования префикса $A[0...i-1]$ в префикс $B[0...j-1]$.

Шаги алгоритма

Инициализируем первую строку матрицы стоимостью вставки символов, умноженной на количество символов; а первый столбец — ценой удаления, соответственно. Основной алгоритм заполняет матрицу слева направо: на каждом шаге выбирается минимум из стоимости трёх операций:

1. Заменить символ ($dp[i-1][j-1] + replace_cost$)
2. Вставить символ ($dp[i][j-1] + insert_cost$)
3. Удалить символ ($dp[i-1][j] + delete_cost$)

Согласно индивидуализации, стоимости считаются в зависимости от символов, стоящих на требуемых местах замены/удаления в строке В/А.

Оценка сложности алгоритма:

Временная сложность

Проход по всем ячейкам матрицы dp .

Общая: $O(|A| |B|)$

Сложность по памяти

Матрица создаётся для всех префиксов входных данных.

Итого: $O(|A| |B|)$

Тестирование

Таблица 1. Тестирование.

Входные данные	Выходные данные
1 1 1 e w w 2 r 3	2
1 1 1 r w w 2 r 3	2
1 1 1 rq w w 2 r 3	3

Вывод

В ходе лабораторной работы были написаны программы с использованием динамического программирования. Также дополнительно был добавлен отладочный вывод для интерпретации результатов работы программы.