

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: Редакционное расстояние

Студент гр. 3388

Лутфулин Д.А.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2025

Цель работы.

Реализовать алгоритм нахождения редакционного расстояния и редакционного предписания между двумя строками.

Задание.

Редакционное расстояние (Вагнер-Фишер)

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

1. $replace(\varepsilon, a, b)$ – заменить символ a на символ b .
2. $insert(\varepsilon, a)$ – вставить в строку символ a (на любую позицию).
3. $delete(\varepsilon, b)$ – 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

Редакционное предписание (Вагнер-Фишер)

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

$replace(\varepsilon, a, b)$ – заменить символ a на символ b .

$insert(\varepsilon, a)$ – 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 .

Пример (все операции стоят одинаково)

М	М	М	Р	І	М	Р	Р
С	О	Н	Н		Е	С	Т
С	О	Н	Е	Н	Е	А	Д

Пример (цена замены 3, остальные операции по 1)

М	М	М	Д	М	І	І	І	І	Д	Д
С	О	Н	Н	Е					С	Т
С	О	Н		Е	Н	Е	А	Д		

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

Выходные данные: первая строка – последовательность операций (М – совпадение, ничего делать не надо; Р – заменить символ на другой; І – вставить символ на текущую позицию; Д – удалить символ из строки); вторая строка – исходная строка A ; третья строка – исходная строка B .

Sample Input:

1 1 1

entrance

reenterable

Sample Output:

IMIMMIMMRRM

entrance

reenterable

Редакционное расстояние (Левенштейн)

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.

Параметры входных данных:

Первая строка входных данных содержит строку из строчных латинских букв. $(SS, 1 \leq |S| \leq 25501 \leq |S| \leq 2550)$.

Вторая строка входных данных содержит строку из строчных латинских букв. $(T, 1 \leq |T| \leq 25501 \leq |T| \leq 2550)$.

Параметры выходных данных:

Одно число LL, равное расстоянию Левенштейна между строками S и T.

Sample Input:

pedestal

stien

Sample Output:

7

Выполнение работы.

Редакционное расстояние(Вагнер-Фишер)

A – первая строка

B – вторая строка

Алгоритм вычисляет минимальную стоимость преобразования одной строки в другую, учитывая цены операции вставки, удаления и замены символов. Для этого создаётся таблица, где каждая ячейка i, j отражает стоимость преобразования префикса $A[:i]$ строки A в префикс $B[:j]$ строки B. В нулевой строке таблицы значения задают стоимость преобразования пустой строки в префикс B, а в нулевом столбце – стоимость преобразования префикса A в пустую строку. Затем таблица заполняется последовательно: если символы совпадают, стоимость остаётся прежней(значение слева сверху от ячейки), иначе выбирается минимальная среди трёх возможных операций с учётом их стоимости. В конце в таблице получается минимальная общая цена редактирования всей строки. Такой метод позволяет эффективно и точно находить оптимальный путь преобразования.

Редакционное предписание:

Для нахождения редакционного предписания мы также строим матрицу расстояний между префиксами строк A и B, но также создаём вторую матрицу, где на соответствующей позиции находится не редакционное расстояние, а символ операции, совершённой для того чтобы попасть в эту ячейку. По этому символу можно понять, из какой ячейки мы попали в данную и тем самым восстановить путь вплоть до самой первой позиции таблицы, с которой мы начинали. Идём с правого нижнего угла до левого верхнего и записываем символ в строку, тогда ответом будет эта строка, выписанная задом наперёд.

Редакционное расстояние(Левентейн)

Частный случай редакционного расстояния Вагнера-Фишера, в котором стоимость всех операций равна 1. Запускаем изначальный алгоритм с данными стоимостями и получаем ответ.

Оценка сложности по времени и памяти.

n – длина строки A

m – длина строки B

Так как строится матрица $m \times n$, а каждый элемент матрицы вычисляется за $O(1)$ (сравнить 2 символа либо 3 числа и 2 символа), сложность алгоритма по времени будет $O(n \times m)$. Помимо матрицы мы храним только 2 строки и переменные, значит сложность по памяти такая же.

В алгоритме нахождения редакционного предписания добавляется используется матрица $m \times n$, где стоят символы операций, а также строка, в которую собираются символы операций. Так как на каждом шаге мы добавляем 1 символ и строковый либо столбцовый индекс обязательно меняется на 1, длина этой строки, а, соответственно и сложность её нахождения, не больше чем $n+m$. Тогда сложность по памяти аналогична

Вывод.

Были разработаны и проанализированы алгоритмы нахождения редакционного расстояния и редакционного предписания между двумя строками.