Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ПРАКТИЧЕСКОЙ РАБОТЕ №12
дисциплины «Алгоритмизация»
Вариант___

Выполнил:
Репкин Александр Павлович
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств вычислительной
техники и автоматизированных систем», очная форма обучения

_____
(подпись)

Руководитель практики:
Воронкин Р.А., канд. техн. наук,
доцент кафедры инфокоммуникаций

_____
(подпись)

Отчет защищен с оценкой _____ Дата защиты_____

Ставрополь, 2023 г.

Порядок выполнения работы:

1. Создана программа на основе приведённых в лекции примеров динамического программирования, для нахождения расстояния Левенштейна (Расстояние Левенштейна – метрика сходства между двумя строковыми последовательностями. Чем больше расстояние, тем более различны строки. Для двух одинаковых последовательностей расстояние равно нулю. По сути, это минимальное число односимвольных преобразований (удаления, вставки или замены), необходимых, чтобы превратить одну последовательность в другую). Приведены три функции: editDistTd и editDistBU, находящие количество изменений, необходимых к внесению, а также reincarnate, на основе матрицы, полученной из editDistBU, анализирующая внесённые изменения.

```
static int editDistTd(int i, int j) {  4 usages
    if (i == 0) return j;
    if (j == 0) return i;
    if (numbers[i][j] == '∞') {
        int ins = editDistTd(i,  j: j - 1) + 1;
        int del = editDistTd( i: i - 1, j) + 1;
        int sub = editDistTd( i: i - 1,  j: j - 1) + (first.charAt(i - 1) != second.charAt(j - 1) ? 1 : 0);
        numbers[i][j] = (char) Integer.min(ins, Integer.min(del, sub));
    }
    return numbers[i][j];
}
```

Рисунок 1. Полученный код функции editDistTd.

```
static int[][] editDistBU() {  1 usage
    int[][] values = new int[needed_n][needed_m];
    for (int i = 0; i < needed_n; i++) values[i][0] = i;
    for (int j = 0; j < needed_m; j++) values[0][j] = j;
    for (int i = 1; i < needed_n; i++)
        for (int j = 1; j < needed_m; j++)
            values[i][j] = Integer.min( a: values[i - 1][j] + 1, Integer.min( a: values[i][j - 1] + 1,  b: values[i - 1][j - 1]
    return values;
}
```

Рисунок 2. Полученный код функции editDistBU.

```
static void reincarnate(int[][] values, String first_input, String second_input) {  1 usage
    StringBuilder first = new StringBuilder();
    StringBuilder second = new StringBuilder();
    int first_length = first_input.length();
    int second_length = second_input.length();
    while (first_length > 0 && second_length > 0) {
        if (values[first_length][second_length] == values[first_length - 1][second_length] + 1) {
            first.insert( offset: 0, first_input.charAt(first_length - 1));
            second.insert( offset: 0,  str: "_");
            first_length--;
        } else if (values[first_length][second_length] == values[first_length][second_length - 1] + 1) {
            first.insert( offset: 0,  str: "_");
            second.insert( offset: 0, second_input.charAt(second_length - 1));
            second_length--;
        } else if (values[first_length][second_length] == values[first_length - 1][second_length - 1] + (first_input.charAt(
            first.insert( offset: 0, first_input.charAt(first_length - 1));
            second.insert( offset: 0, second_input.charAt(second_length - 1));
            first_length--;
            second_length--;
        }
    }
    System.out.println("Using data from editDistBU, program has found changes:");
    System.out.println("Changes in first word: " + first);
    System.out.println("Changes in second word: " + second);
}
```

Рисунок 3. Полученный код функции reincarnate.

```
Good day! Please, enter your words:
First: автор
Great! And now second: авторство
Minimum Edit Distance, according to editDistTD: 4
Minimum Edit Distance, according to editDistBU: 4
Using data from editDistBU, program has found changes:
Changes in first word: автор____
Changes in second word: авторство
```

Рисунок 4. Пример выполнения программы.

Вывод: в ходе выполнения практической работы были рассмотрены примеры динамического программирования на задаче нахождения расстояния Левенштейна. В процессе выполнения работы проведено ознакомление с динамическим программированием, чья суть заключается в решении сложных задач путём разбиения их на более простые подзадачи.