

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 ЛАБОРАТОРНОЙ РАБОТЕ №12**  
**дисциплины «Алгоритмизация»**

Выполнил:  
Магдаев Даламбек Магомедович  
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., доцент кафедры  
инфокоммуникаций

---

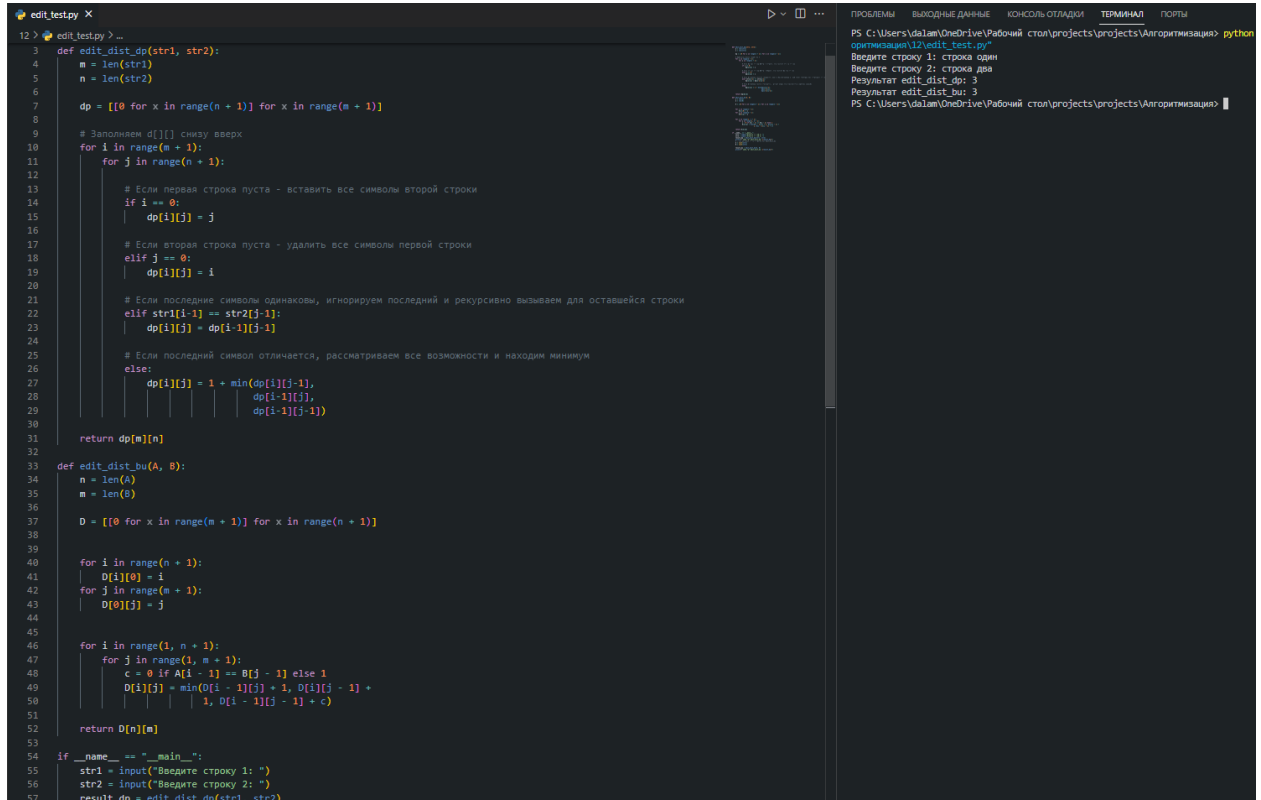
(подпись)

Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

Ставрополь, 2023 г.

## Порядок выполнения работы:

1. Написал программу поиска расстояния редактирования с использованием динамического программирования снизу вверх и сверху вниз:



```
12 > edit_testpy > ...
3 def edit_dist_dp(str1, str2):
4     m = len(str1)
5     n = len(str2)
6     dp = [[0 for x in range(n + 1)] for x in range(m + 1)]
7
8     # Заполняем d[][] снизу вверх
9     for i in range(m + 1):
10         for j in range(n + 1):
11             # Если первая строка пуста - вставить все символы второй строки
12             if i == 0:
13                 dp[i][j] = j
14
15             # Если вторая строка пуста - удалить все символы первой строки
16             elif j == 0:
17                 dp[i][j] = i
18
19             # Если последние символы одинаковы, игнорируем последний и рекурсивно вызываем для оставшейся строки
20             elif str1[i-1] == str2[j-1]:
21                 dp[i][j] = dp[i-1][j-1]
22
23             # Если последний символ отличается, рассматриваем все возможности и находим минимум
24             else:
25                 dp[i][j] = 1 + min(dp[i][j-1],
26                                   dp[i-1][j],
27                                   dp[i-1][j-1])
28
29     return dp[m][n]
30
31
32
33 def edit_dist_bu(A, B):
34     n = len(A)
35     m = len(B)
36
37     D = [[0 for x in range(m + 1)] for x in range(n + 1)]
38
39
40     for i in range(n + 1):
41         D[i][0] = i
42     for j in range(m + 1):
43         D[0][j] = j
44
45     for i in range(1, n + 1):
46         for j in range(1, m + 1):
47             c = 0 if A[i-1] == B[j-1] else 1
48             D[i][j] = min(D[i-1][j] + 1, D[i][j-1] + 1,
49                           D[i-1][j-1] + c)
50
51     return D[n][m]
52
53
54 if __name__ == "__main__":
55     str1 = input("Введите строку 1: ")
56     str2 = input("Введите строку 2: ")
57     result_dp = edit_dist_dp(str1, str2)
```

ПРОБЛЕМЫ Выходные данные КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ python

PS C:\Users\vdalam\OneDrive\Рабочий стол\projects\projects\Алгоритмизация> python edit\_test.py

Введите строку 1: строка один

Введите строку 2: строка два

Результат edit\_dist\_dp: 3

Результат edit\_dist\_bu: 3

PS C:\Users\vdalam\OneDrive\Рабочий стол\projects\projects\Алгоритмизация>

Рисунок 1. Код программы edit\_dist.py и результат выполнения

**Вывод:** в результате выполнения лабораторной работы были изучены алгоритмы heap sort поиска расстояния редактирования с использованием динамического программирования снизу вверх и сверху вниз.