

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»**

Выполнил:

Лейс Алексей Вячеславович
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»

(подпись)

Руководитель практики кандидат тех.
наук доцент кафедры
инфокоммуникаций: Воронкин Р.А

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Порядок выполнения работы:

Решения задачи поиска расстояния редактирования и восстановления решения используя динамическое программирование как снизу-вверх, так и сверху-вниз

```
1 using System;
2 using System.Collections.Generic;
3
4 namespace
5 {
6     class Program
7     {
8         static void Main()
9         {
10             string str1 = "editing";
11             string str2 = "distance";
12
13             var result = EditDist(str1, str2);
14
15             foreach (var item in result.Item2)
16             {
17                 Console.WriteLine($"{item.Item1} {item.Item2}");
18             }
19         }
20
21         static (int, List<char, char>) EditDist(string a, string b)
22         {
23             int[,] matrix = new int[a.Length + 1, b.Length + 1];
24
25             int EditDistTd(int i, int j)
26             {
27                 if (matrix[i, j] == 0)
28                 {
29                     if (i == 0)
30                     {
31                         matrix[i, j] = j;
32                     }
33                     else if (j == 0)
34                     {
35                         matrix[i, j] = i;
36                     }
37                     else
38                     {
39                         int ins = EditDistTd(i, j - 1) + 1;
40                         int delete = EditDistTd(i - 1, j) + 1;
41                         int sub = EditDistTd(i - 1, j - 1) + (a[i - 1] != b[j - 1] ? 1 : 0);
42                         matrix[i, j] = Math.Min(Math.Min(ins, delete), sub);
43                     }
44                 }
45                 return matrix[i, j];
46             }
47
48             void EditDistBu()
49             {
50                 for (int i = 0; i <= a.Length; i++)
51                 {
52                     matrix[i, 0] = i;
53                 }
54                 for (int j = 1; j <= b.Length; j++)
55                 {
56                     matrix[0, j] = j;
57                 }
58                 for (int i = 1; i <= a.Length; i++)
59                 {
60                     for (int j = 1; j <= b.Length; j++)
61                     {
62                         int c = a[i - 1] != b[j - 1] ? 1 : 0;
63                         matrix[i, j] = Math.Min(Math.Min(matrix[i - 1, j] + 1, matrix[i, j - 1] + 1), matrix[i - 1, j - 1] + c);
64                     }
65                 }
66             }
67
68             List<char, char> Restore()
69             {
70                 List<char, char> result = new List<char, char>();
71                 int i = a.Length, j = b.Length;
72                 while (i != 0 || j != 0)
73                 {
74                     if (i != 0 && matrix[i, j] == matrix[i - 1, j] + 1)
75                     {
76                         result.Add((a[i - 1], '-'));
77                         i--;
78                     }
79                     else if (j != 0 && matrix[i, j] == matrix[i, j - 1] + 1)
80                     {
81                         result.Add('-', b[j - 1]);
82                         j--;
83                     }
84                     else if (matrix[i, j] == matrix[i - 1, j - 1] + (a[i - 1] != b[j - 1] ? 1 : 0))
85                     {
86                         result.Add((a[i - 1], b[j - 1]));
87                         i--;
88                         j--;
89                     }
90                 }
91                 result.Reverse();
92                 return result;
93             }
94
95             EditDistTd(a.Length, b.Length);
96             EditDistBu();
97             var solution = Restore();
98
99             return (matrix[a.Length, b.Length], solution);
100         }
101     }
102 }
```