

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего
образования

АДЫГЕЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Инженерно-физический факультет

Кафедра автоматизированных систем обработки информации и управления

Отчет по практике
Дистанция Левенштейна
2 курс, группа 2ИВТ2

Выполнил:

_____ Д.А. Толокнов

«___» _____ 2023г.

Руководитель:

_____ С.В. Теплоухов

«___» _____ 2023г.

Майкоп, 2023 г.

Введение

1. Задание
2. Ход работы
 - (a) Пример кода
 - (b) Формулы для задачи
 - (c) Изображение решения
3. Литература

1.Задание

Дистанция Левенштейна.

2.Ход работы

а. Пример кода

```
#include <iostream>
#include<string>
#include<Windows.h>
using namespace std;

int main()
{
    SetConsoleCP(1251);//для ввода кириллицы
    SetConsoleOutputCP(1251);//для вывода кириллицы
    string a;//первое слово
    string b;//второе слово
    cout << "введите слова ПЕРВОЕ _ ВТОРОЕ:";

    //ввод слов
    cin>> a >> b;

    //размер слов
    int aVal = a.size();
    int bVal = b.size();
    // cout << aVal << " _ " << bVal << endl; //вывод размера слов

    //создание двумерного массива
    int** p = new int* [aVal + 1];
    for (int i = 0; i < aVal + 1; i++) {
        p[i] = new int[bVal + 1];
    }
    //заполнение первой строки и первого столбца
    for (int i = 0; i <= aVal; i++) {
        p[i][0] = i;
    }
    for (int i = 0; i <= bVal; i++) {
        p[0][i] = i;
    }

    // составление таблицы по условиям
    int va;
    int vb;
    int vc;
    int vd;
    for (int j = 1; j <= bVal; j++) {
        for (int i = 1; i <= aVal; i++) {
            va = p[i][j - 1] + 1;
            vb = p[i - 1][j] + 1;
            if (a[i - 1] == b[j - 1]) {
                vd = 0;
            }
        }
    }
}
```

```

    }
    else {
        vd = 1;
    }
    vc = p[i - 1][j - 1] + vd;

    if (va <= vb && va <= vc) {
        p[i][j] = va;
    }
    else if (vb <= va && vb <= vc) {
        p[i][j] = vb;
    }
    else if (vc <= va && vc <= vb) {
        p[i][j] = vc;
    }
};
}

//вывод таблицы
for (int i = 0; i <= bVal; i++) {
    for (int j = 0; j <= aVal; j++) {
        cout << p[j][i] << " ";
    }
    cout << endl;
}

cout << "ответ: понадобится " << p[aVal][bVal] << " шагов, чтобы превратить " << b <

//удаление массива
for (int i = 0; i < aVal + 1; i++) {
    delete p[i];
}
delete p;
}

```

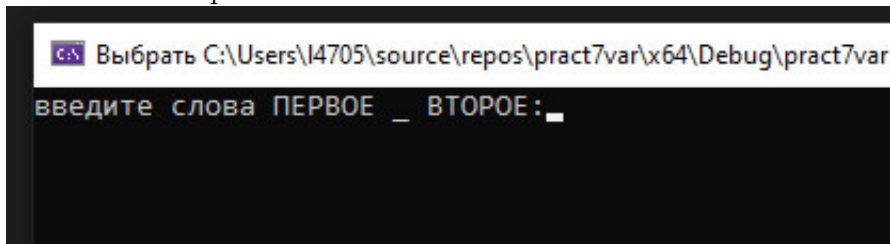
б. Формула для задачи

Вычислим расстояние между двумя строками методом Вагнера — Фишера: составим матрицу D и каждый её элемент вычислим по рекуррентной формуле:

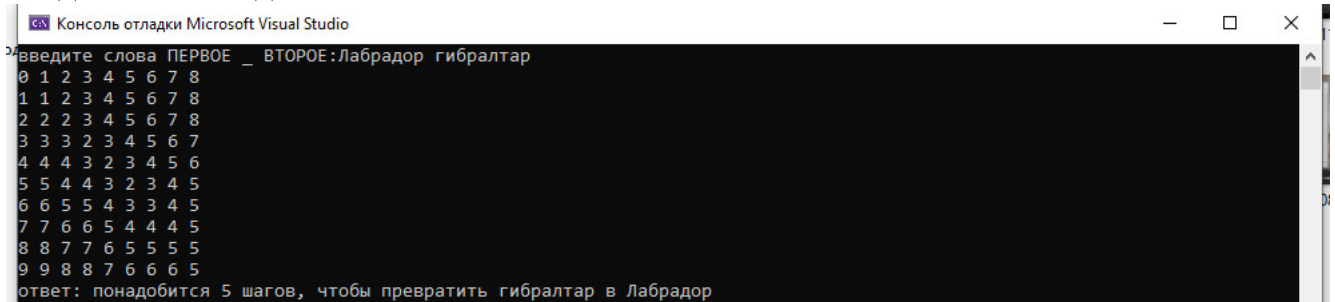
$$D(i, j) = \begin{cases} 0 & ; i = 0, j = 0 \\ i * deleteCost & ; j = 0, i > 0 \\ j * insertCost & ; i = 0, j > 0 \\ D(i - 1, j - 1) & ; S_1[i] = S_2[j] \\ \min (& \\ \quad D(i, j - 1) + insertCost & \\ \quad D(i - 1, j) + deleteCost & ; j > 0, i > 0, S_1[i] \neq S_2[j] \\ \quad D(i - 1, j - 1) + replaceCost & \\) & \end{cases}$$

с. Изображение решения

Начальный экран



Ввод слов и вывод ответа



Литература

1. <https://habr.com/ru/articles/676858/>