**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторным работам №2

«**Расстояние Левенштейна**»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-31Б |  | преподаватель каф. ИУ5 |
| Баринов А. А. |  | Гапанюк Ю. Е. |
| Подпись и дата: |  | Подпись и дата: |
|  |  |  |

Москва, 2023 г.

Задание:

Найти расстояние Левенштейна для двух словТекст программы

Текст программы

.py

def levenshtein\_distance(str1, str2):

len\_str1 = len(str1) + 1

len\_str2 = len(str2) + 1

matrix = [[0 for n in range(len\_str2)] for m in range(len\_str1)]

for i in range(len\_str1):

matrix[i][0] = i

for j in range(len\_str2):

matrix[0][j] = j

for i in range(1, len\_str1):

for j in range(1, len\_str2):

if str1[i - 1] == str2[j - 1]:

S = 0

else:

S = 1

matrix[i][j] = min(

matrix[i - 1][j] + 1,

matrix[i][j - 1] + 1,

matrix[i - 1][j - 1] + S

)

return matrix[len\_str1 - 1][len\_str2 - 1]

word1 = "топот"

word2 = "грохот"

distance = levenshtein\_distance(word1, word2)

print(f"Расстояние Левенштейна между '{word1}' и '{word2}': {distance}")

.c

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int minim(int a, int b, int c) {

if (a < b && a < c) {

return a;

}

else if (b < c) {

return b;

}

else {

return c;

}

}

int levenshtein\_distance(const char\* str1, const char\* str2) {

int len\_str1 = strlen(str1) + 1;

int len\_str2 = strlen(str2) + 1;

int\*\* matrix = (int\*\*)malloc(len\_str1 \* sizeof(int\*));

for (int i = 0; i < len\_str1; i++) {

matrix[i] = (int\*)malloc(len\_str2 \* sizeof(int));

}

for (int i = 0; i < len\_str1; i++) {

matrix[i][0] = i;

}

for (int j = 0; j < len\_str2; j++) {

matrix[0][j] = j;

}

int S;

for (int i = 1; i < len\_str1; i++) {

for (int j = 1; j < len\_str2; j++) {

if (str1[i - 1] == str2[j - 1])

S = 0;

else S = 1;

matrix[i][j] = minim(

matrix[i - 1][j] + 1,

matrix[i][j - 1] + 1,

matrix[i - 1][j - 1] + S

);

}

}

int result = matrix[len\_str1 - 1][len\_str2 - 1];

for (int i = 0; i < len\_str1; i++) {

free(matrix[i]);

}

free(matrix);

return result;

}

int main() {

const char\* word1 = "Medic";

const char\* word2 = "Adict";

int distance = levenshtein\_distance(word1, word2);

printf("Levenshtein distance between '%s' and '%s': %d\n", word1, word2, distance);

return 0;

}

Результат выполнения

Расстояние Левенштейна между 'топот' и 'грохот': 3