Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

Пенза 2021

**ОТЧЕТ**

по лабораторной работе №6

по курсу «Логика и основы алгоритмизации в инженерных задачах»

на тему «Поиск расстояний в графе»

**Выполнили:**

Студенты группы 20ВВ2:

Зиновьев Я.М.

Аверочкин П.С.

**Приняли:**

к. т. н. доцент Юрова О.В.

д. т. н., профессор Митрохин М.А.

### **Название:**

Поиск расстояний в графе.

### **Задание 1**

1. Сгенерируйте (используя генератор случайных чисел) матрицу смежности для неориентированного графа *G*. Выведите матрицу на экран.
2. Для сгенерированного графа осуществите процедуру поиска расстояний, реализованную в соответствии с приведенным выше описанием. При реализации алгоритма в качестве очереди используйте класс **queue** из стандартной библиотеки С++.

**3.**\* Реализуйте процедуру поиска расстояний для графа, представленного списками смежности.

### **Цель работы:**

Написать функции и разработать программу по заданию.

**Описание программы:**

Программа осуществляет поиск расстояний между вершинами графа. Реализована данная функция с помощью обхода в ширину.  
Создается массив, который представляет нам матрицу смежности. При обходе в ширину кроме посещенных вершин так же записываются и расстояния.

### **Листинг:**

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <iostream>

#include <locale.h>

#include <malloc.h>

#include <queue>

using namespace std;

void BFSD(int\*\* G, int n, int num, int \*dist)

{

    queue <int> q;

    int i;

    dist[num] = 0;

    q.push(num);

    while (!q.empty())

    {

        num = q.front();

        q.pop();

        cout << num + 1 << " ";

        for (i = 0; i < n; i++)

        {

            if (dist[i] == -1 && G[num][i] == 1)

            {

                q.push(i);

                dist[i] = dist[num] + 1;

            }

        }

    }

    cout << endl;

}

void InitDist(int\* mass, int n)

{

    for (int i = 0; i < n; i++)

    {

        mass[i] = -1;

    }

}

void PrintMass(int n, int\*\* mass)

{

    cout << endl;

    for (int i = 0; i < n; i++)

    {

        for (int j = 0; j < n; j++)

        {

            cout << mass[i][j] << " ";

        }

        cout << endl;

    }

}

void InitMatrix(int\*\* matrix, int n)

{

    for (int i = 0; i < n; i++)

    {

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

        for (int j = 0; j < n; j++)

        {

            matrix[i][j] = rand() % 2;

            if (i == j)

            {

                matrix[i][j] = 0;

            }

        }

    }

    for (int i = 0; i < n; i++)

    {

        for (int j = 0; j < n; j++)

        {

            matrix[i][j] = matrix[j][i];

        }

    }

}

int main(void)

{

    setlocale(LC\_ALL, "russian");

    setvbuf(stdin, NULL, \_IONBF, 0);

    setvbuf(stdout, NULL, \_IONBF, 0);

    int \*\* G = NULL, \* dist, n, num;

    srand(time(NULL));

    cout << "Введите размерность матрицы: " << endl;

    cin >> n;

    system("cls");

    dist = (int\*)malloc(sizeof(int) \* n);

    InitDist(dist, n);

    G = (int\*\*)malloc(n \* sizeof(int));

    InitMatrix(G, n);

    cout << "Матрица:";

    PrintMass(n, G);

    cout << "Введите номер начальной вершины: \n";

    cin >> num;

    cout << "Проход и расстояние:\n";

    BFSD(G, n, num - 1, dist);

    for (int i = 0; i < n; i++)

    {

        cout << dist[i] << " ";

    }

    free(G);

}

### **Результаты работы программы:**

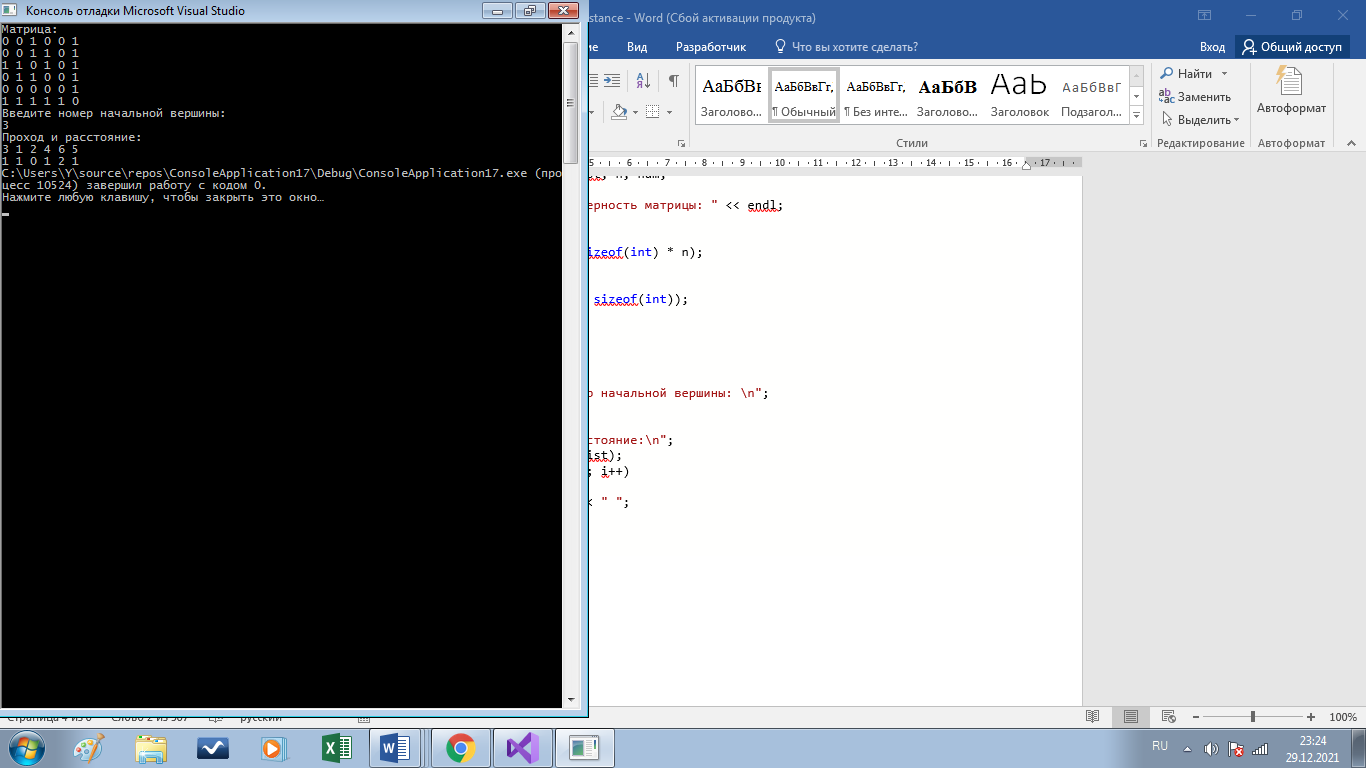


Рисунок 1

### **Вывод:** В ходе выполнения данной лабораторной работы мы осуществили поиск расстояний между вершинами графа и реализовали данную функция с помощью обхода в ширину.