Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №7

по курсу «Логика и основы алгоритмизации в инженерных задачах»

на тему «Поиск расстояний во взвешенном графе»

Выполнили:

студенты группы 19ВВ4

Каравайкин И.М.

Савельев А.А.

Приняли:

Митрохин М.А.

Юрова О.В.

Пенза 2020

### Цель работы

Изучение алгоритма поиска длины пути в графе

### Лабораторное задание

1. Сгенерируйте (используя генератор случайных чисел) матрицу

смежности для неориентированного взвешенного 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++.

### Листинг

#define \_CRT\_SECURE\_NO\_WARNINGS

#define \_CRT\_SECURE\_NO\_WARNING

#include "stdio.h"

#include <stdlib.h>

#include "conio.h"

#include "malloc.h"

#include <locale>

#include <queue>

using namespace std;

void bfsd(int v, int\*\* arr, int\* dist, int m) {

queue <int> q;

q.push(v);

dist[v] = 0;

while (!q.empty()) {

v = q.front();

q.pop();

printf("%d, ", v);

for (int j = 0; j < m; j++) {

if ((arr[v][j] != 0) && (dist[j] == -1)) {

q.push(j);

dist[j] = dist[v] + arr[v][j];

}

}

}

}

void main() {

srand(time(NULL));

setlocale(LC\_ALL, "Russian");

int n;

printf("Введите размер массива: ");

scanf("%d", &n);

int\*\* arr1 = (int\*\*)malloc(n \* sizeof(int\*));

for (int i = 0; i < n; i++) {

arr1[i] = (int\*)malloc(n \* sizeof(int));

}

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

for (int i = 0; i < n; i++) {

for (int j = 0; j < n; j++) {

arr1[i][j] = rand() % 2;

if (arr1[i][j] == 1) {

arr1[i][j] = rand() % 5;

}

}

}

for (int i = 0; i < n; i++) {

for (int j = 0; j < n; j++) {

arr1[j][i] = arr1[i][j];

if (i == j) { arr1[i][j] = 0; }

printf("%d ", (arr1[i][j]));

}

printf("\n");

}

printf("\n\n\n");

for (int i = 1; i < n; i++) {

dista[i] = -1;

}

bfsd(0, arr1, dista, n);

printf("\n\n\n");

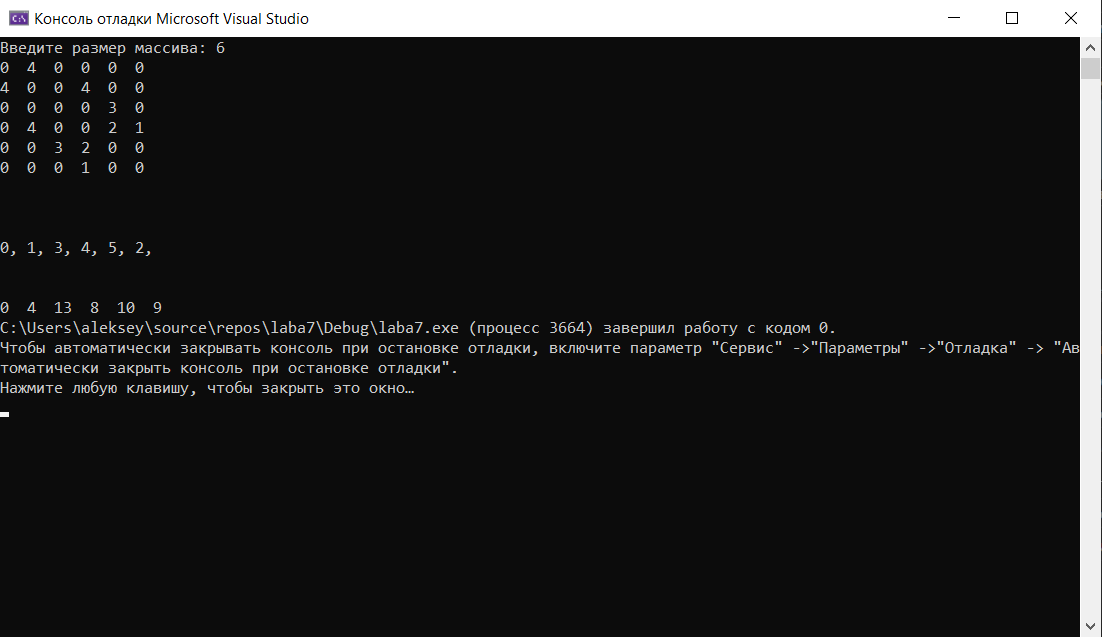
for (int i = 0; i < n; i++) {

printf("%d ", dista[i]);

}

}

### Результат работы



Длина пути от заданной вершины до n-ой вершины

- Порядок обхода графа в ширину

- Случайно сгенерированная матрица

Рисунок 1 – Результат работы

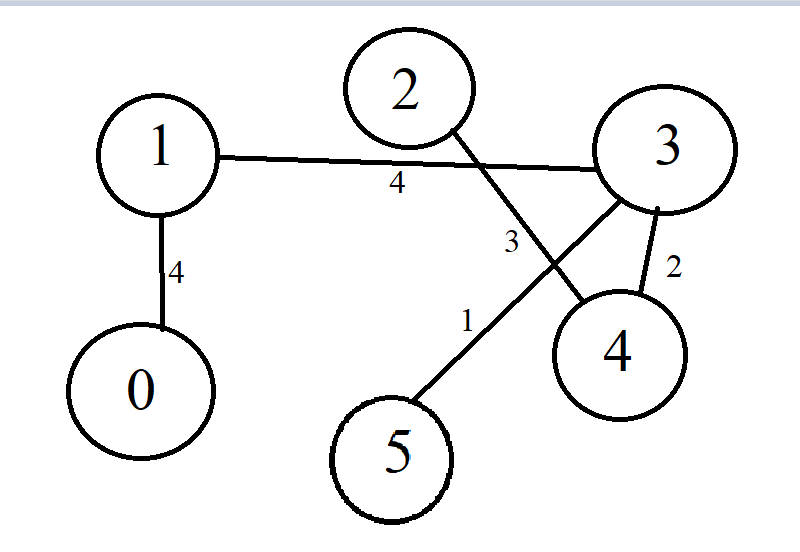


Рисунок 2 – Граф по сгенерированной матрице смежности

### Выводы

В ходе выполнения лабораторной работы была разработана программа, осуществляющая поиск расстояний между вершинами во взвешенном графе.