Пензенский государственный университет

Кафедра «Вычислительной техники»

**Отчёт**

По лабораторной №7

по курсу «Логика и основы алгоритмизации в инженерных задачах»

на тему «Поиск расстояний во взвешенном графе»

**Выполнил студент группы 20ВВ1:**

Будников М.С.

**Приняли:**

Юрова О.В.

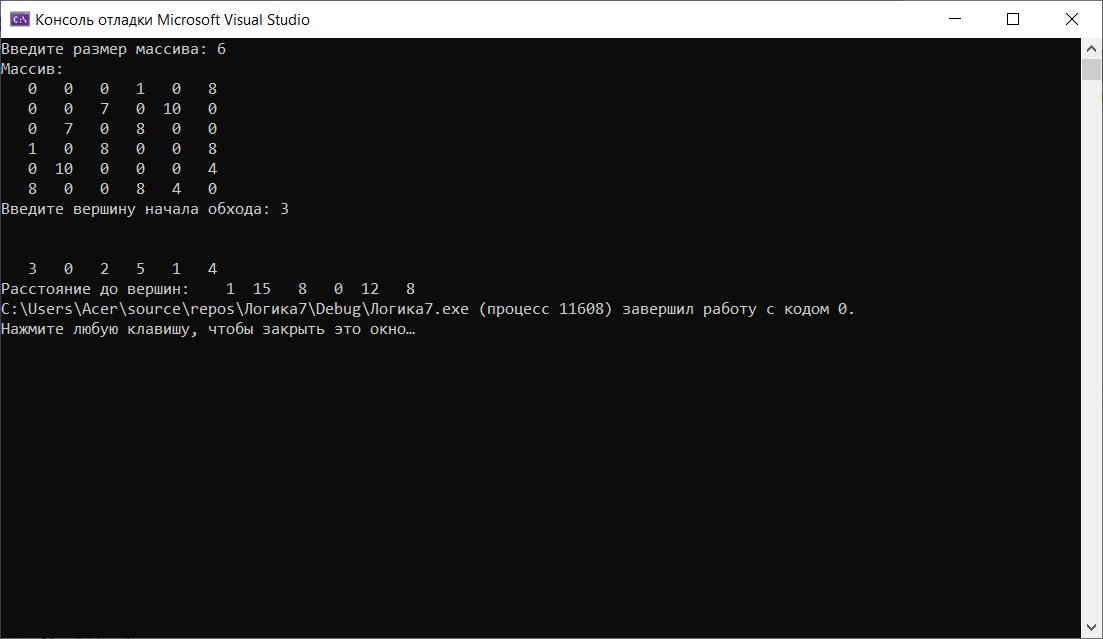
Акифьев И.В.

Пенза 2021 г.

**Цель работы:** освоить алгоритм поиска расстояний во взвешенном графе.

**Задание 1:**

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++.



1. \*Сгенерируйте (используя генератор случайных чисел) матрицу смежности для ориентированного взвешенного графа G. Выведите матрицу на экран и осуществите процедуру поиска расстояний, реализованную в соответствии с приведенным выше описанием.

**Задание 2\*:**

1. Модернизируйте программу так, чтобы получить возможность запуска программы с параметрами командной строки (см. описание ниже). В качестве параметра должны указываться тип графа (взвешенный или нет) и наличие ориентации его ребер (есть ориентация или нет).

**Листинг:**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <locale.h>

#include <queue>

using namespace std;

int\* vis;

int num;

int\*\* m;

void BFSD(int num);

void BFSD(int c) {

int s;

queue<int> q;

vis[c] = 0;

q.push(c);

while (!q.empty()) {

s = q.front();

q.pop();

printf("%4i", s);

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

if (m[s][i] > 0 && vis[i] > (m[s][i] + vis[s])) {

q.push(i);

vis[i] = m[s][i] + vis[s];

}

}

}

printf("\nРасстояние до вершин: ");

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

printf("%4i", vis[i]);

}

}

int main()

{

setlocale(LC\_ALL, "Rus");

int generate;

int c;

printf("Введите размер массива: ");

scanf("%i", &num);

m = new int\* [num];

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

m[i] = new int[num];

}

srand(time(NULL));

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

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

if (j == i) {

m[i][j] = 0;

}

else if (j > i) {

generate = rand() % 100;

if (generate >= 50) {

m[i][j] = rand() % 11;

}

else {

m[i][j] = 0;

}

}

else {

m[i][j] = m[j][i];

}

}

}

//вывод массива

printf("Массив:");

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

printf("\n");

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

printf("%4i", m[i][j]);

}

}

vis = new int[num];

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

vis[i] = 10000;

}

printf("\nВведите вершину начала обхода: ");

scanf("%i", &c);

printf("\n");

printf("\n");

BFSD(c);

}

**Вывод:** в процессе выполнения лабораторной работы были получены навыки реализации алгоритма поиска расстояний во взвешенном графе.