Министерство науки и высшего образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

ОТЧЁТ

по лабораторной работе №9

по курсу «Логика и основы алгоритмизации в инженерных задачах»

на тему «Поиск расстояний в графе»

Выполнил:

студент группы 21ВВ2

Шилов Алексей

Приняли:

д.т.н. Митрохин М.А.

к.т.н. Юрова О.В.

Пенза 2022

Задание 1

Сгенерировать (используя генератор случайных чисел) матрицу смежности для неориентированного графа G. Вывести матрицу на экран.

Задание 2

Для сгенерированного графа осуществить процедуру поиска расстояний. При реализации алгоритма в качестве очереди использовать класс queue из стандартной библиотеки C++.

Листинг

#include <iostream>

#include <queue>

using namespace std;

const int N = 5;

int\*\* M = (int\*\*)malloc(N \* sizeof(int));

int Dist[N];

void BFSD(int s)

{

queue <int> Q;

Q.push(s);

Dist[s] = 0;

while (!Q.empty())

{

s = Q.front();

Q.pop();

cout << s + 1 << endl;

for (int i = 0; i < N; i++)

{

if (M[s][i] == 1 and Dist[i] == -1) {

Q.push(i);

Dist[i] = Dist[s] + 1;

}

}

}

}

int main()

{

srand(time(NULL));

for (int i = 0; i < N; i++)

{

M[i] = (int\*)malloc(N \* sizeof(int));

Dist[i] = -1;

}

for (int i = 0; i < N; i++) //Заполнение

{

for (int j = 0; j < N; j++)

{

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

M[j][i] = M[i][j];

if (i == j)

M[i][j] = 0;

}

}

for (int i = 0; i < N; i++) //Вывод

{

for (int j = 0; j < N; j++)

{

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

}

cout << "\n";

}

cout << "\n";

system("pause");

for (int i = 0; i < N; i++)

{

if (Dist[i] == -1) {

BFSD(i);

}

}

cout << endl;

for (int i = 0; i < N; i++)

{

cout << Dist[i] << endl;

}

free(M);

}

Результаты



Рисунок 1

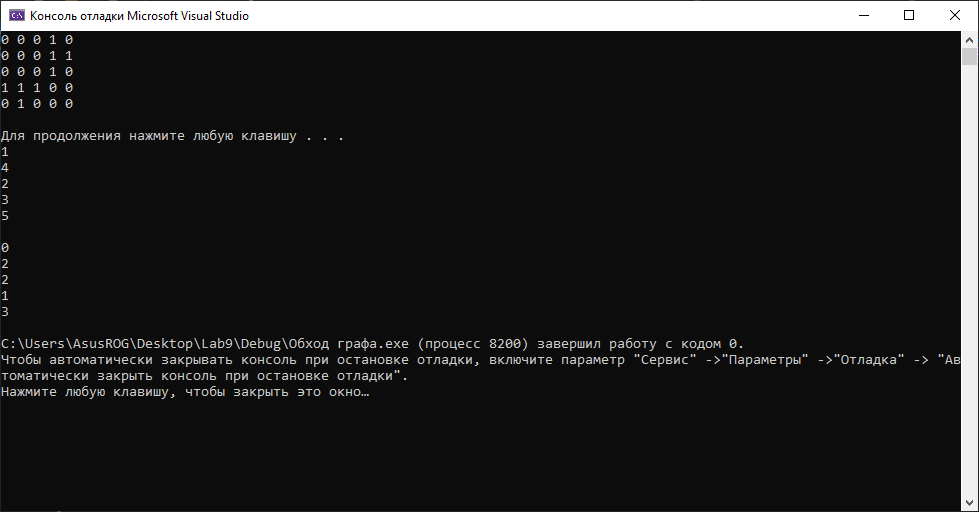


Рисунок 2

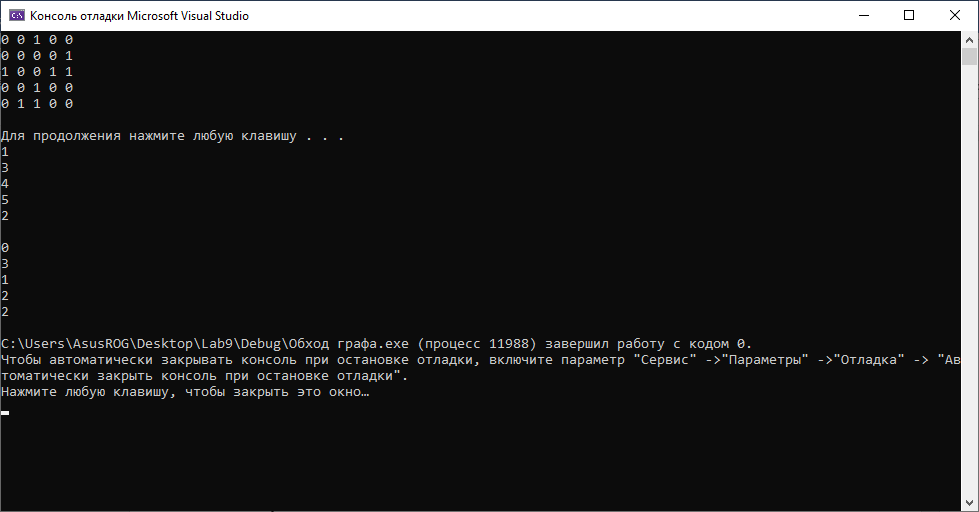


Рисунок 3

Вывод

Изучил поиск расстояний в графе, изучил методы реализации поиска расстояний в графе.