Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №5

по курсу «Логика и основы алгоротмизирования в ИС»

на тему «Обход графа в ширину»

Выполнили

студенты группы 20ВВ2:

Тумасов Вадим

Портнов Никита

Принял:

д.т.н Митрохин М.А.

д.т.н Юрова О.В.

Пенза 2021

**Лабораторное задание:**

Задание 1

1. Сгенерируйте (используя генератор случайных чисел) матрицу

смежности для неориентированного графа G. Выведите матрицу на экран.

2. Для сгенерированного графа осуществите процедуру обхода в

глубину, реализованную в соответствии с приведенным выше описанием.

.

**Листинг:**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <ctime> // Библиотека времени

#include <cmath>

#include <locale.h>

#include <iostream> // Библиотека Ввода,Вывода

#include <conio.h>

#include < queue >

int\*\* matrix;

using namespace std;

void bfs(int v, int n)

{

int p = 0, i = 0, u = 0;

char\* NUM;

NUM = new char[n];

for (u = 0; u < n; u++)

{

NUM[u] = 0;

}

cout « v - 1;

//i = s;

queue <int> plan; // план посещения в виде очереди

plan.push(v - 1); // мы нумеруем с 0, а не с 1

NUM[v - 1] = 1; // отмечаем, что эта вершина уже заносилась в план

//int counter = 1; // начальную уже сосчитали

while (!plan.empty()) {

v = plan.front(); // посещаем следующую по плану вершину

plan.pop(); // удаляем ее из плана посещения

//cout « v;

for (int i = 0; i < n; i++) { // перебираем соседние с ней

if ((matrix[v][i] == 1) && (NUM[i] == 0)) { // если новая, то

plan.push(i); // добавляем ее в план

cout « i;

NUM[i] = 1; // отмечаем, что уже не новая

}

}

}

}

int main()

{

int p = 0, i = 0, n;

int v = 0;

//int s;

setlocale(LC\_ALL, "Russian"); // подключение русского

srand(time(NULL)); // новый отсчёт времени

cout « "n="; // выводит данные

cin » n; // помещает данные

matrix = new int\* [n];

//cout « "n=";

//cin » n;

cout « "Матрица: \n";

int z;

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

{

matrix[i] = new int[n];

}

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

{

for (p = 0; p < n; p++)

{

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

}

}

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

{

for (p = 0; p < n; p++)

{

if (p == i)

{

matrix[i][p] = 0;

}

else

{

matrix[i][p] = matrix[p][i];

}

}

}

cout « "\n";

cout « "\n";

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

{

// used[i] = false;

for (p = 0; p < n; p++)

cout « matrix[i][p] « " "; // выводит матрицу

cout « endl;

}

cout « "Стартовая вершина » ";

//cin » s;

cin » v;

//bool\* vis = new bool[n];

cout « "Порядок обхода: ";

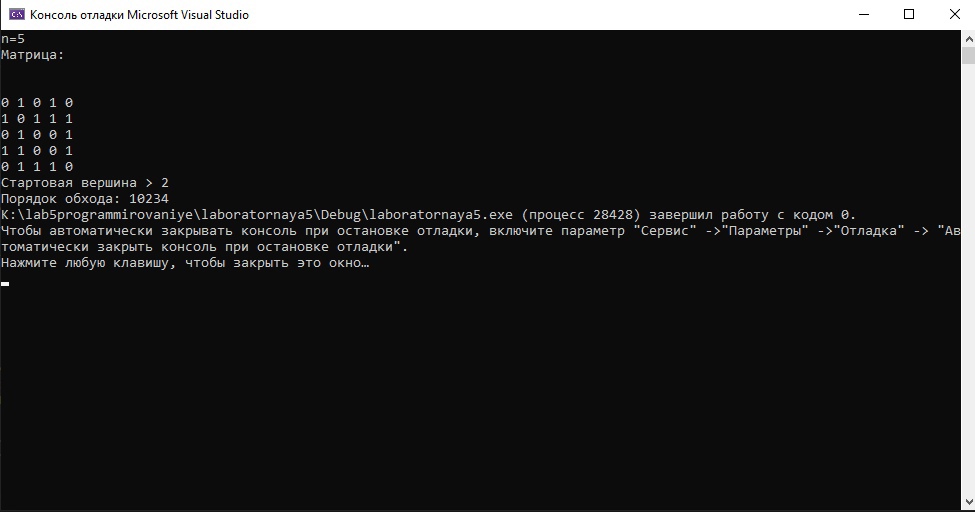
bfs(v, n);

//delete[] used;

\_getch();

}

**Результат работы программы:**



**Вывод**: Мы сгенерировали матрицу смежности для неориентированного графа, организовали обход в ширину помощью bfs и класса queue.