Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

## по дисциплине "Логика и основы алгоритмизации в инженерных задачах"

## по лабораторной работе №2

## на тему «Обход графа в ширину»

Выполнил студент гр. 22ВВС1:

Беспалов Н.А.

Приняли:

Юрова О.В.

Акифьев И.В.

Пенза 2024

**Цель работы**: Реализовать алгоритм обхода графа в ширину

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 ширину,

### реализованную в соответствии с приведенным выше описанием. При

### реализации алгоритма в качестве очереди используйте класс queue из

### стандартной библиотеки С++.

### **Листинг**

#define SRT\_SECURE\_NO\_WORNINGS

#include <iostream>

#include <stdio.h>

#include <queue>

#include <stdlib.h>

#include <time.h>

using namespace std;

void bfs(int v, int\*\* p, int n, int\* vis)

{

queue <int> q{};

q.push(v);

vis[v] = true;

while (!q.empty())

{

v = q.front();

q.pop();

cout << ">> " << v + 1 << " ";

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

{

if (p[v][i] == 1 and vis[i] == false)

{

q.push(i);

vis[i] = true;

}

}

}

}

void vivod(int\*\* p, int n)

{

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

{

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

{

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

}

cout << endl;

}

}

int main()

{

setlocale(LC\_ALL, "Rus");

int i, j, n, v; //n - размер матрицы, v - вершина начала обхода алгоритма

cout << "Введите размер матрицы ";

cin >> n;

cout << "Введите вершину для начала поиска в ширину ";

cin >> v;

v -= 1;

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

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

{

vis[i] = 0;

}

int\*\* p = (int\*\*)malloc(n \* sizeof(int\*)); //выделение памяти для матрицы смежности

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

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

}

srand(time(NULL)); //заполнение массива p

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

{

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

{

if (i == j)

{

p[i][j] = 0;

}

else

{

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

}

}

}

for (i = 0; i < n; i++) //отзеркаливание матрицы

{

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

{

p[j][i] = p[i][j];

}

}

vivod(p, n);

cout << endl << endl;

bfs(v, p, n, vis);

free(p);

free(vis);

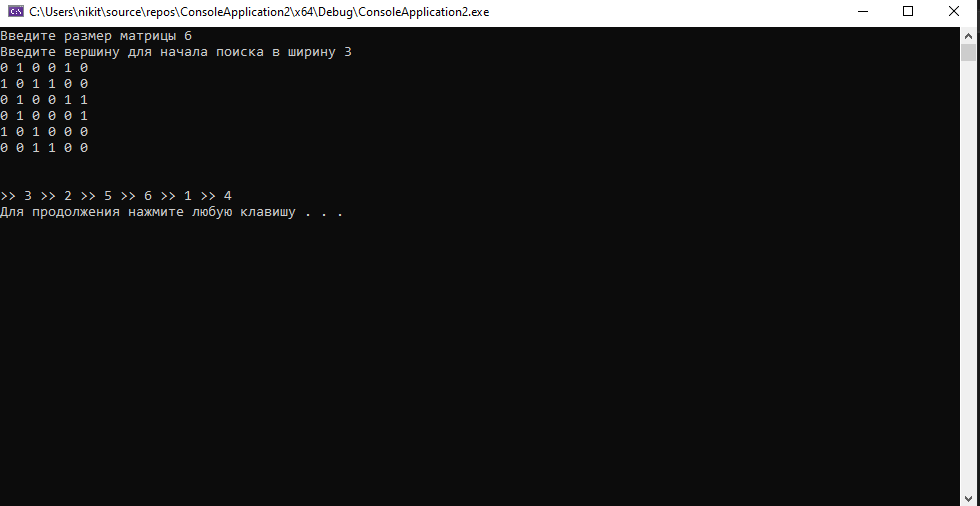
cout << endl;

system("pause");

return 0;

}

**Результаты работы программы**



## Вывод

Освоили алгоритм обхода в ширину. Научились реализовывать его с помощью стандартной библиотеки <queue>.