Пензенский государственный университет

Кафедра "Вычислительной техники"

**Отчёт**

По лабораторной №5

по дисциплине "Л и ОА в ИТ"

на тему "Обход графа в ширину"

***Выполнили студенты группы 19вв1:***

*Грозов В.*

***Приняли:***

*Митрохин М. А.*

*Юрова О. В.*

Пенза 2020

# Цель работы: выполнить обход 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++.

**Теоретическая часть:**

Обход графа в ширину – еще один распространенный способ обхода

графов.

Основная идея такого обхода состоит в том, чтобы посещать вершины

по уровням удаленности от исходной вершины. Удалённость в данном

случае понимается как количество ребер, по которым необходимо прейти до

достижения вершины.

Класс Queue представляет обычную очередь, работающую по алгоритму FIFO ("первый вошел - первый вышел").

У класса Queue можно отметить следующие методы:

* Dequeue: извлекает и возвращает первый элемент очереди
* Enqueue: добавляет элемент в конец очереди
* Peek: просто возвращает первый элемент из начала очереди без его удаления

**Практическая часть:**

**Листинг:**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <stdio.h>

#include <locale>

#include <malloc.h>

#include <stdlib.h>

#include <time.h>

#include <conio.h>

#include <stdbool.h>

#include <queue>

void BFS(int p, int arr);

int \*\*mass, \*visit;

void BFS(int p, int arr) {

std::queue <int> Q;

Q.push(p);

visit[p] = 1;

while (!Q.empty()) {

p = Q.front();

printf("%d ", p);

Q.pop();

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

if (mass[p][i] == 1 && visit[i] == false) {

Q.push(i);

visit[i] = 1;

}

}

}

}

int main()

{

setlocale(0, "rus");

int p, k;

int arr;

printf("Введите матрицу:");

scanf\_s("%d", &arr);

mass = (int\*\*)malloc(arr\*sizeof(int));

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

mass[i] = (int \*)malloc(arr\*sizeof(int));

}

visit = (int\*)malloc(arr\*sizeof(int));

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

visit[i] = 0;

}

printf("Введем матрицу:\n");

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

for (int j = 0;j<i + 1;j++) {

k = rand() % 2;

if (i != j) {

mass[i][j] = k;

mass[j][i] = k;

}

else {

mass[i][j] = 0;

}

}

}

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

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

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

}

printf("\n");

}

printf("\n");

for (int i = 0;i<1;i++) {

printf("Выберете вершину с которой необходимо начанете обход:");

scanf\_s("%d", &p);

if (p <= arr) {

printf("\nВы проходите");

}

else {

printf("Выберете новую координату\n");

i--;

}

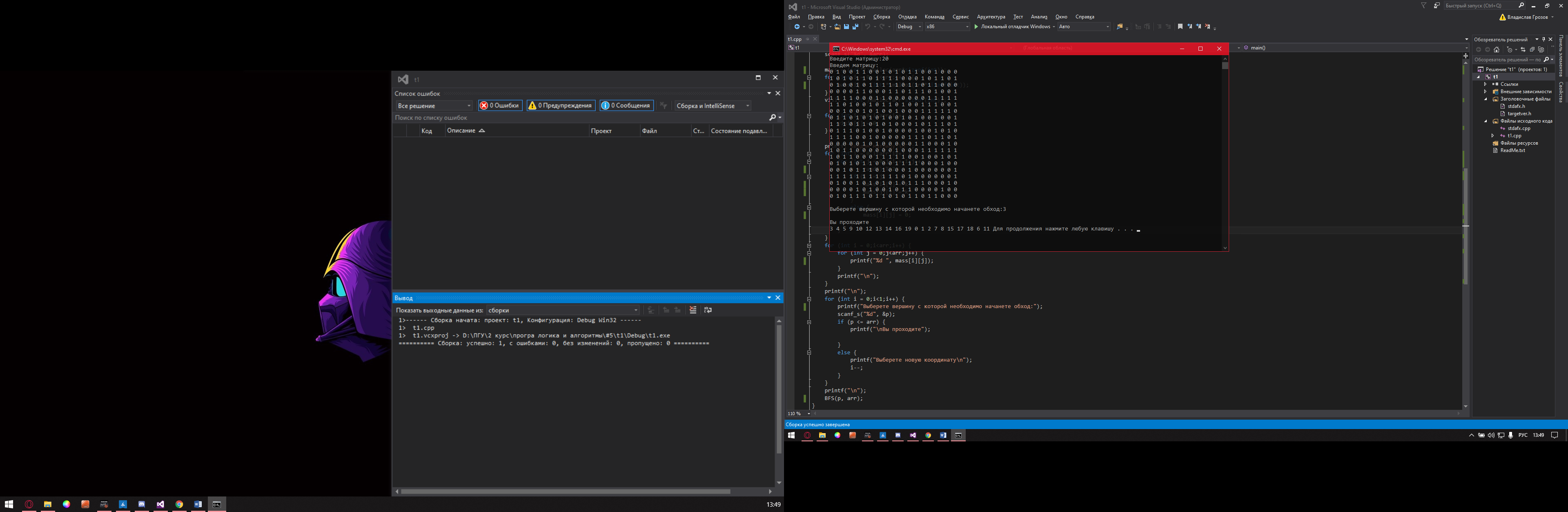
}

printf("\n");

BFS(p, arr);

}

**Результат работы программы:**



**Вывод:** Научились выполнять обход графа в ширину.