Министерство образования Российской Федерации   
Пензенский государственный университет   
Кафедра «Вычислительная техника»

ОТЧЕТ

По лабораторной №5   
На тему: «Обход графа в ширину»  
По дисциплине: «Логика и основы алгоритмизации в инженерных задачах»

Выполнили студенты группы 20ВВ2:

Борунов Д.А

Сурков М.С

Приняли:

Митрохин М.А. Юрова О.В.

**Пенза 2020г**

**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++.

**3.\*** Реализуйте процедуру обхода в ширину для графа, представленного

списками смежности.

**Задание 2\***

**1.** Для матричной формы представления графов реализуйте алгоритм

обхода в ширину с использованием очереди, построенной на основе

структуры данных «список», самостоятельно созданной в лабораторной

работе № 3.

**2.** Оцените время работы двух реализаций алгоритмов обхода в ширину

(использующего стандартный класс queue и использующего очередь,

реализованную самостоятельно) для графов разных порядков.

Метод выполнения задания

**Задание 1.1 и 1.2.**

Создать матрицу смежности графа (двумерный динамический массив). Описать функцию:

DFS(int st, int\*\* b, int num\_of\_elem, bool\* visited)

int st **–** стартовая вершина

int\*\* b – указатель на сгенерированную матрицу смежности

bool\* visited – указатель на массив посещенных вершин

int num\_of\_elem – количество элементов в сгенерированной матрице

Перед вызовом функции пометить все вершины сгенерированного графа как не помеченные (False) и с читать с клавиатуры стартовой вершины графа для последующей передачи ее в функцию.

В теле функции DFS создать очередь Q. Вложить с помощью команды push() стартовую вершину в созданную очередь и пометить ее, как посещенную. Далее описать цикл, который будет выполняться при условии, что очередь НЕ пуста (empty()). В теле цикла отметить текущую вершину, вивести ее на экран и удалить из очереди. После этого вложить цикл, который будет проходить по строкам сгенерированной матрицы смежности и добавлять в очередь, вершины, которые смежны текущей и не посещены, а затем помечать их как посещенные.

**Листинг программы**

#include <iostream>

#include <stdio.h>

#include <string.h>

#include <cstdlib>

#include <windows.h>

#include <conio.h>

#include <ctime>

#include <cmath>

#include <stack>

#include <queue>

using namespace std;

int max\_size = 6, i, j, start;

bool\* vektor = (bool\*)malloc(max\_size \* sizeof(bool));

void DFS(int st, int\*\* b, int num\_of\_elem, bool\* visited) {

queue<int> Q;

int t;

Q.push(st);

visited[st] = true;

while (!Q.empty())

{

t = Q.front();

Q.pop();

printf("%d ", t + 1);

for (int i = 0; i < num\_of\_elem; i++)

{

if (b[t][i] == 1 && !visited[i])

{

Q.push(i);

visited[i] = true;

}

}

}

}

int main() {

setlocale(LC\_ALL, "Rus");

srand(time(NULL));

int\*\* a = new int\* [max\_size];

for (i = 0; i < max\_size; i++)

{

a[i] = new int[max\_size];

}

for (i = 0; i < max\_size; i++)

{

for (j = i + 1; j < max\_size; j++)

{

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

a[j][i] = a[i][j];

}

}

for (i = 0; i < max\_size; i++)

{

vektor[i] = false;

for (j = 0; j < max\_size; j++)

{

if (i == j)

a[i][j] = 0;

printf("%2d", a[i][j]);

}

printf("\n");

}

///////////////////////////////////////

printf("\nОбход графа в ширину\ncтартовая вершина - ");

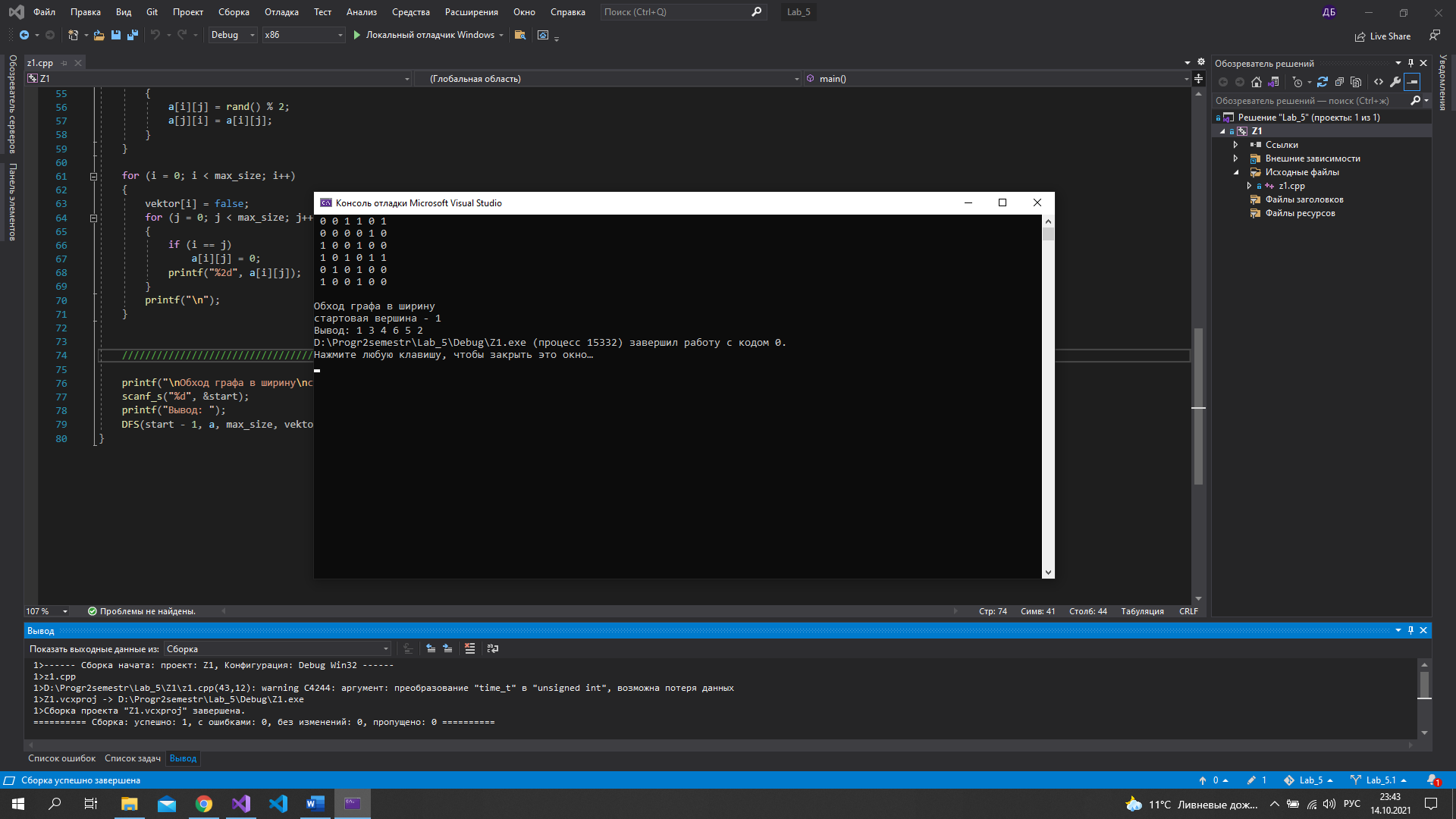
scanf\_s("%d", &start);

printf("Вывод: ");

DFS(start - 1, a, max\_size, vektor);

}

**Результат работы программы**



**Вывод:** в ходе выполнения лабораторной реализовали алгоритм обхода графа в ширину несколькими способами