Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №8

по дисциплине: «Логика и основы алгоритмизации в инженерных задачах.»

на тему: «Обход графа в ширину.»

Выполнили:

студенты группы 21ВВ4

Гребенев А.С.

Никишин Д.Д.

Принял:

Акифьев И. В.  
Юрова О. В.

Пенза, 2022

**Цель работы:** реализовать программу, которая выполняет обход графа в ширину.

**Лабораторная работа:**

**Задание 1**

1. Сгенерируйте (используя генератор случайных чисел) матрицу

смежности для неориентированного графа G. Выведите матрицу на экран.

2. Для сгенерированного графа осуществите процедуру поиска

расстояний, реализованную в соответствии с приведенным выше описанием.

При реализации алгоритма в качестве очереди используйте класс queue из

стандартной библиотеки С++.

Листинг:

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <stdio.h>

#include <conio.h>

#include <locale.h>

#include <malloc.h>

#include <stdlib.h>

#include <time.h>

#include <queue>

using namespace std;

int\*\* G;

int\* vis;

int ver;

void BFS(int v, int n)

{

queue<int> q; // объявление

q.push(v); // добавления в очередь вершину

vis[v] = 1;

while (!q.empty())

{

v = q.front();

printf(" %d", v);

q.pop();

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

{

if (G[v][i] == 1 && vis[i] == 0)

{

vis[i] = 1;

q.push(i);

}

}

}

}

int main()

{

int n;

n = 0;

int k = 0;

setlocale(LC\_ALL, "RUS");

printf("Введите размерность матрицы: ");

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

srand(time(NULL));

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

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

{

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

}

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

{

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

{

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

if (i == j)

{

G[i][j] = 0;

}

G[j][i] = G[i][j];

}

}

printf("\nМатрица смежности:\n");

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

{

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

{

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

}

printf("\n");

}

printf("\n");

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

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

{

vis[i] = 0;

}

printf("Введите вершину, с которой нужно начинать обход графа в ширину (0..4): ");

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

printf("\nРезультат: ");

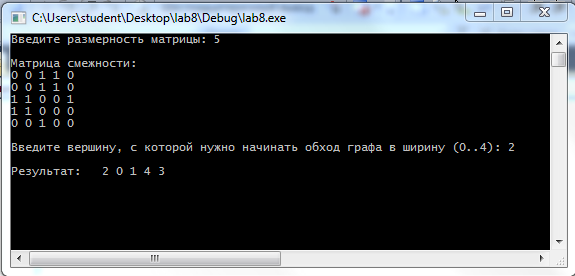
BFS(ver, n);

getchar ();

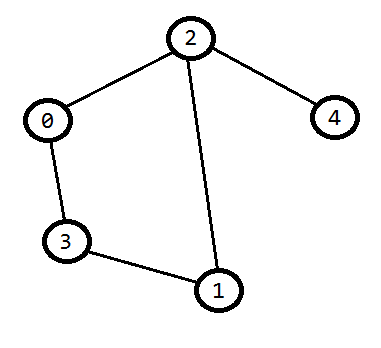
getchar ();

}

Результат работы программы:



Проверка:



**Вывод:** мы реализовали обход графа в ширину, используя список смежности и функцию queue.