Пензенский государственный университет  
Кафедра «Вычислительной техники»

**Отчет**по лабораторной работе №5  
по дисциплине: «Логика и основы алгоритмизации в инженерных задачах»  
на тему: «Обход графа в ширину»

**Выполнил студент группы 19ВВ1:**

Тебнев Р.  
Балалаев А.

**Приняли:**

д.т.н. профессор Митрохин М. А.

Юрова О.В.

Пенза 2020

**Цель работы:** изучить обход в глубину, научится реализовывать его

**Теория**

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. Например, если для графа на рисунке 1 начать обход из первой вершины, то вершины 3, 6 и 2 будут находиться на уровне удаленности в 1 ребро, а вершины 5 и 4 на уровне удаленности в 2 ребра.

Вход: G – матрица смежности графа.

Выход: номера вершин в порядке их прохождения на экране.

Алгоритм ПОШ

1.1. для всех i положим NUM[i] = False пометим как &quot;не посещенную&quot;;

1.2. ПОКА существует &quot;новая&quot; вершина v

1.3. ВЫПОЛНЯТЬ BFS (v).

Алгоритм BFS(v):

2.1. Создать пустую очередь Q = {};

2.2. Поместить v в очередь Q.push(v);

2.3. пометить v как &quot;посещенную&quot; NUM[v] = True;

2.4. ПОКА Q != Æ очередь не пуста ВЫПОЛНЯТЬ

2.5. v = Q.front() установить текущую вершину;

2.6. Удалить первый элемент из очереди Q.pop();

2.7. вывести на экран v;

2.8. ДЛЯ i = 1 ДО size\_G ВЫПОЛНЯТЬ

2.9. ЕСЛИ G(v,i) = = 1И NUM[i] = = False

2.10. ТО

2.11. Поместить i в очередь Q.push(i);

2.12. пометить v как &quot;посещенную&quot; NUM[v] = True;

**Код программы**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <time.h>

#include <locale.h>

#include <queue>

using namespace std;

#define N 5

void newMat(int\* mas) {

srand(time(0));

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

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

if (i == j) {

mas[N \* i + j] = 0;

}

if (i < j) {

mas[N \* i + j] = rand() % 2;

mas[j \* N + i] = mas[N \* i + j];

}

}

}

}

void printMat(int\* mas) {

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

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

printf("%d ", mas[N \* i + j]);

}

printf("\n");

}

}

void BFS(int s, int\* G, int\* num) {

queue <int> q;

q.push(s);

num[s] = 1;

while (!q.empty()) {

s = q.front();

printf("%d ", s);

q.pop();

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

if (G[s \* N + i] == 1 && num[i] == 0) {

q.push(i);

num[i] = 1;

}

}

}

}

int main() {

int G[N][N], num[N], s;

setlocale(LC\_ALL, "rus");

newMat(&G[0][0]);

printMat(&G[0][0]);

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

num[i] = 0;

}

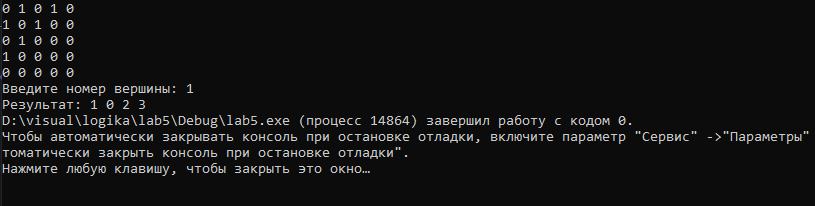
printf("Введите номер вершины: ");

scanf("%d", &s);

printf("Результат: ");

}

**Результат программы**



**Вывод:** изучили обход графа в ширину и научились реализовывать его.