Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №5

по курсу «Логика и основы алгоритмизации в инженерных задачах»

на тему «Обход графа в ширину»

Выполнил:

студент группы 20ВВ2

Кочергин В.П.

Принял:

к.т.н. доцент Юрова О.В.

д.т.н. профессор Митрохин М. А.

Пенза 2021

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

**Задание 1** Сгенерировать матрицу и список смежности. Осуществить процедуру обхода в ширину

Листинг:

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <windows.h>

#include <queue>

using namespace std;

struct Node

{

int items;

Node\* next;

};

typedef Node\* PNode;

PNode\* Head;

void InitializHead(int size)

{

int i;

Head = (PNode\*)malloc(size \* sizeof(PNode));

for (i = 0; i < size; i++) Head[i] = NULL;

}

PNode CreateNode(int items)

{

PNode NewNode = (Node\*)malloc(sizeof(Node));

NewNode->items = items;

NewNode->next = NULL;

return NewNode;

}

void AddFirst(PNode NewNode, int i)

{

NewNode->next = Head[i];

Head[i] = NewNode;

}

void AddAfter(PNode q, PNode NewNode)

{

NewNode->next = q->next;

q->next = NewNode;

}

void AddLast(PNode NewNode, int i)

{

PNode q = Head[i];

while (q->next) q = q->next;

AddAfter(q, NewNode);

}

void print(int i)

{

PNode q = Head[i];

while (q)

{

printf("%d ", q->items);

q = q->next;

}

printf("\n");

}

void BFS(int \*vis, int \*\*a, int size, int num)

{

queue <int> q;

int i;

q.push(num);

vis[num] = 1;

while (!q.empty())

{

num = q.front();

q.pop();

printf("%d ", num);

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

{

if (vis[i] == 0 && a[num][i] == 1)

{

q.push(i);

vis[i] = 1;

}

}

}

}

void BFS2(int\* vis, int\*\* a, int size, int num)

{

queue <int> q;

int i;

q.push(num);

vis[num] = 1;

while (!q.empty())

{

num = q.front();

q.pop();

printf("%d ", num);

while (Head[num])

{

if (vis[Head[num]->items] == 0)

{

q.push(Head[num]->items);

vis[Head[num]->items] = 1;

}

Head[num] = Head[num]->next;

}

}

}

int main()

{

int i, j, size, \*\* a, \* vis, ran, num;

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

setvbuf(stdin, NULL, \_IONBF, 0);

setvbuf(stdout, NULL, \_IONBF, 0);

PNode NewNode = NULL;

printf("Введите размеры массива ");

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

printf("Введите начальную строку ");

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

printf("Матрица смежности:\n");

a = (int\*\*)malloc(size \* sizeof(int));

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

srand(time(NULL));

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

{

vis[i] = 0;

}

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

{

a[i] = (int\*)malloc(size \* sizeof(int));

}

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

{

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

{

a[i][j] = 0;

}

}

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

{

for (j = 1 + i; j < size; j++)

{

ran = rand() % 101;

if (ran > 30) a[i][j] = 1;

else a[i][j] = 0;

}

}

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

{

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

{

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

}

}

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

{

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

{

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

}

printf("\n");

}

printf("\nОчерёдность посещения вершин: ");

BFS(vis, a, size, num);

InitializHead(size);

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

{

NewNode = CreateNode(i);

AddFirst(NewNode, i);

}

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

{

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

{

if (a[i][j] == 1)

{

NewNode = CreateNode(j);

AddLast(NewNode, i);

}

}

}

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

{

vis[i] = 0;

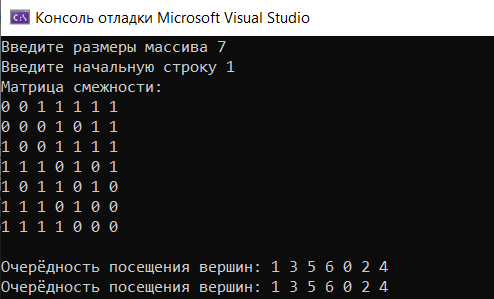
}

printf("\nОчерёдность посещения вершин: ");

BFS2(vis, a, size, num);

}

Результат работы программы



**Вывод:** В конечном результате удалось написать алгоритм для обхода графа в ширину