|  |
| --- |
| Министерство образования Российской Федерации  Пензенский государственный университет  Кафедра «Вычислительная техника» |
| Отчет  по лабораторной работе №4  по курсу «Логика и основы алгоритмизации в инженерных задачах»  на тему «Обход графа в глубину» |
|  |
|  |
| Выполнила студентка группы 19ВВ3:  Ханбекова Е. В.  Принял:  Митрохин М. А. |
| Пенза  2020 |

### **Задание 1**

1. Сгенерируйте (используя генератор случайных чисел) две матрицу смежности для неориентированного графа *G*. Выведите сгенерированные матрицы на экран.
2. Для сгенерированного графа осуществите процедуру обхода в глубину, реализованную в соответствии с приведенным выше описанием.

### **Задание 2\***

1. Для матричной формы представления графов выполните преобразование рекурсивной реализации обхода графа к не рекурсивной.

### **Листинг**

#include <stdio.h>

#include <conio.h>

#include <Windows.h>

#include <stdlib.h>

#include <stack>

#include <iostream>

using namespace std;

void OGVG(int \*\*A, int N, int i, int\*M) {

int j;

if (M[i] != 1) {

M[i] = 1;

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

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

if (A[i][j] == 1) {

OGVG(A, N, j, M);

}

}

}

}

void OVGBR(int \*\*A, int N, int\*M) {

int j, i;

stack <int> S;

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

if (M[i] != 1) {

M[i] = 1;

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

S.push(i);

do {

i = S.top();

j = 0;

while (j < N) {

if ((A[i][j] == 1) && (M[j] != 1)) {

S.push(j);

M[j] = 1;

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

i = j;

j = 0;

}

else

j++;

}

S.pop();

} while (S.empty() != true);

}

}

}

int main(void)

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int N, i, j;

printf("Ведите размер матрицы \n");

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

srand(6);

int \*M = (int\*)malloc(N \* sizeof(int));

int \*\*A = (int \*\*)malloc(N \* sizeof(int \*));

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

A[i] = (int \*)malloc(N \* sizeof(int));

}

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

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

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

A[j][i] = A[i][j];

if (i == j) {

A[i][j] = 0;

}

}

}

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

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

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

}

printf("\n");

}

printf("\nОбход в глубину рекурсией:\n");

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

OGVG(A, N, i, M);

}

printf("\n");

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

M[i] =0;

}

printf("\nОбход в глубину без рекурсии:\n");

OVGBR(A, N, M);

free(A);

free(M);

\_getch();

}

### **Результат работы программы**

