Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №7

# по курсу «Логика и основы алгоритмизации в инженерных задачах»

на тему «Обход графа в глубину»

**Выполнили**

студенты группы 21ВВ3

Чинов Даниил

Тюкалов Василий

**Приняли:**

Юрова О.В.

Митрохин М.А.

Пенза 2022

# Название

Обход графа в глубину

**Цель работы** – изучение алгоритма обхода графа в глубину.

# 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

глубину, реализованную в соответствии с приведенным выше описанием.

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

списками смежности.

**Задание 2\***

1. Для матричной формы представления графов выполните

преобразование рекурсивной реализации обхода графа к не рекурсивной.

# Листинг

Задание 1

//Выполнили студенты группы 21ВВ3: Тюкалов В.Е. и Чинов Д.Д.

#include<iostream>

#include<stdio.h>

#include<stdlib.h>

#include<time.h>

#include<math.h>

#include<windows.h>

typedef struct Node {

int data = 0;

Node\* next = NULL;

};

void DFS(int\*\* G, int\* numG, int v, int total) {

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

if (G[v][i] == 1) {

if (numG[v] == 1) {

numG[v] = 0;

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

}

if (numG[i] == 1) DFS(G, numG, i, total);

}

}

}

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int total = 0, v = 0, count = 0;

Node\* Cell = NULL, \* sw = NULL;

printf("Введите размер матрицы: ");

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

int\* numG = (int\*)malloc(total \* sizeof(int));

int\* vPerv = (int\*)malloc(total \* sizeof(int));

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

vPerv[i] = NULL;

}

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

Node\*\* lastG = (Node\*\*)malloc(total \* sizeof(Node\*));

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

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

lastG[i] = (Node\*)malloc(total \* sizeof(Node));

numG[i] = 1;

}

srand(time(NULL));

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

for (int j = i; j < total; j++) {

if (i == j) {

G[i][j] = 0;

}

else {

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

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

}

}

}

printf("\n Матрица смежности \n");

printf("\n ");

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

printf(" %c", (i + 97));

}

printf("\n ");

for (int i = 0; i < total \* 3; i++) {

printf("\_");

}

printf("\n");

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

printf(" %c | ", (i + 97));

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

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

}

printf("\n");

}

printf("\n Список смежности \n");

printf("\n ");

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

Cell = NULL;

lastG[i] = NULL;

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

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

Node\* newCell = (Node\*)malloc(sizeof(Node));

newCell->data = j;

newCell->next = NULL;

if (Cell != NULL) Cell->next = newCell;

else lastG[i] = newCell;

Cell = newCell;

}

}

}

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

Node\* sw = (Node\*)malloc(sizeof(Node));

sw = lastG[i];

printf("%c:", i + 97);

while (sw != NULL) {

printf(" %c", sw->data + 97);

sw = sw->next;

}

free(sw);

printf("\n ");

}

printf("\n");

while (v < total) {

sw = lastG[v];

if (sw == NULL) v++;

while (sw != NULL) {

if (numG[sw->data] == 1) {

printf("%c ", v + 97);

numG[v] = 0;

v = sw->data;

break;

}

sw = sw->next;

if (sw == NULL && numG[v] == 1) {

printf("%c ", v + 97);

numG[v] = 0;

}

if (sw == NULL) v++;

}

}

printf("\n");

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

numG[i] = 1;

}

v = 0;

while (v < total) {

DFS(G, numG, v, total);

v++;

}

}

Задание 2

//Выполнили студенты группы 21ВВ3: Тюкалов В.Е. и Чинов Д.Д.

#include<iostream>

#include<stdio.h>

#include<stdlib.h>

#include<time.h>

#include<math.h>

#include<windows.h>

typedef struct Node {

int data = 0;

Node\* next = NULL;

};

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int total = 0, v = 0, count = 1, countPerv = -1, countV = 1, clearV = 0;

printf("Введите размер матрицы: ");

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

int\* numG = (int\*)malloc(total \* sizeof(int));

int\* vPerv = (int\*)malloc(total \* sizeof(int));

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

vPerv[i] = NULL;

}

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

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

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

numG[i] = 1;

}

srand(time(NULL));

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

for (int j = i; j < total; j++) {

if (i == j) {

G[i][j] = 0;

}

else {

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

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

}

}

}

printf("\n Матрица смежности \n");

printf("\n ");

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

printf(" %c", (i + 97));

}

printf("\n ");

for (int i = 0; i < total \* 3; i++) {

printf("\_");

}

printf("\n");

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

printf(" %c | ", (i + 97));

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

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

}

printf("\n");

}

while (count == 1 && countV == 1) {

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

if (G[v][i] == 1 && numG[i] == 1) {

if (numG[v] == 1) printf("%d ", (v + 1));

numG[v] = 0;

countPerv++;

vPerv[countPerv] = v;

v = i;

break;

}

if (i == total - 1) {

if (numG[v] == 1 && countPerv != -1) {

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

numG[v] = 0;

//break;

}

if (countPerv == -1) v++;

if (countPerv >= 0) {

v = vPerv[countPerv];

countPerv--;

if (countPerv == -1) clearV = 1;

break;

}

if (clearV == 1) countV = 0;

}

}

count = 0;

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

if (numG[i] == 1) count = 1;

}

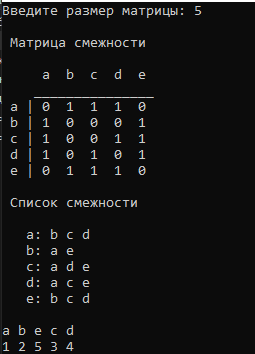
}

}

# Результат работы программы

**Задание 1**

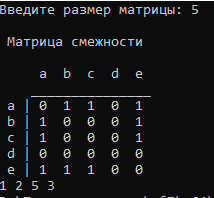
Результаты работы программы показаны на рисунке 1.



**Рисунок 1 – Результат работы программы**

**Задание 2**

Результаты работы программы показаны на рисунке 2.



**Рисунок 2 – Результат работы программы**

# Вывод

# В ходе выполнения лабораторной работы была разработана программа, в который был реализован алгоритм обхода графа в глубину.