Министерство науки и высшего образования

Пензенский государственный университет

Кафедра «Вычислительная техника»

Отчет

По лабораторной работе №5

по курсу «Логика и основы алгоритмизации в инженерных задачах»

на тему «Определение характеристик графов»

Выполнил студент группы 21ВВ2:

Асянов И. К.

Приняли:

Юрова О.В.

Митрохин М.А.

Пенза 2022

Ход работы:

Сгенерировали с помощью генератора случайных чисел матрицу  
смежности для неориентированного графа. Матрица выводится на экран. Осуществили процедуры по определению размера графа, а также поиска его изолированных, концевых и доминирующих вершин.

Листинг:

#include <iostream>

#include <windows.h>

using namespace std;

bool\* visits;

bool\*\* adjacencyMatrix;

int coun;

void Generatorrand(bool\*\* \_matrix, int \_SIZE)

{

for (int row = 0; row < \_SIZE; ++row)

{

\_matrix[row] = new bool[\_SIZE];

}

for (int row = 0; row < \_SIZE; ++row)

{

for (int col = 0; col < \_SIZE; ++col)

{

\_matrix[row][col] = 0;

}

}

bool buffer = 0;

for (int row = 0; row < \_SIZE; ++row)

{

for (int col = 0; col < \_SIZE; ++col)

{

buffer = rand() % 2;

\_matrix[row][col] = buffer;

\_matrix[col][row] = buffer;

}

}

for (int counter = 0; counter < \_SIZE; ++counter)

{

\_matrix[counter][counter] = 0;

}

}

void amount(bool\*\* \_matrix, int \_SIZE) {

for (int row1 = 0; row1 < \_SIZE; ++row1)

{

for (int col1 = 0; col1 < \_SIZE; ++col1)

{

if ((row1 < col1) && \_matrix[row1][col1] == 1) {

coun++;

}

}

}

cout <<"Размер графа: " << coun;

}

void top(bool\*\* \_matrix, int \_SIZE) {

int po = 0;

int um1 = 0;

int um2 = 0;

int um3 = 0;

bool flag1 = true;

bool flag2 = true;

bool flag3 = true;

int row1;

for (int col1 = 0; col1 < \_SIZE; ++col1)

{

po = 0;

for (row1 = 0; row1 < \_SIZE; ++row1)

{

po += \_matrix[row1][col1];

}

if (po == 0) {

if (flag1) {

cout << endl << "Изолированные вершины: ";

flag1 = false;

}

cout <<col1<<" ";

um1++;

}

if (po == 1) {

if (flag2) {

cout << endl<<"Концевые вершины: ";

flag2 = false;

}

cout << col1 << " ";

um2++;

}

if (po == (\_SIZE - 1)) {

if (flag3) {

cout << endl<<"Доминирующие вершины: ";

flag3 = false;

}

cout << col1 << " ";

um3++;

}

}

if (flag1) {

cout << endl << "Изолированные вершины отсутствуют"<<endl;

}

if (flag2) {

cout <<endl <<"Концевые вершины отсутствуют"<<endl;

}

if (flag3) {

cout << endl << "Доминирующие вершины отсутствуют"<<endl;

}

}

void Output(bool\*\* \_matrix, int \_SIZE)

{

for (int row = 0; row < \_SIZE; ++row)

{

for (int col = 0; col < \_SIZE; ++col)

{

cout << \_matrix[row][col] << ' ';

}

cout << endl;

}

}

void Zero(bool\* \_matrix, int \_SIZE)

{

for (int counter = 0; counter < \_SIZE; ++counter)

{

\_matrix[counter] = 0;

}

}

int main()

{

srand(time(NULL));

setlocale(0, "rus");

cout << "Введите размер" << endl;

int SIZE;

cin >> SIZE;

bool\*\* adjacencyMatrix = new bool\* [SIZE];

bool\* visits = new bool[SIZE];

//Zero(visits, SIZE);

Generatorrand(adjacencyMatrix, SIZE);

Output(adjacencyMatrix, SIZE);

amount(adjacencyMatrix, SIZE);

top(adjacencyMatrix, SIZE);

delete[] adjacencyMatrix;

delete[] visits;

}

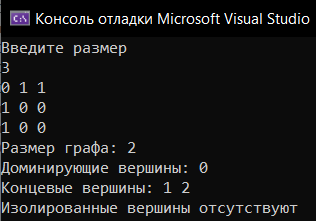


Рис.1 - Результат работы программы

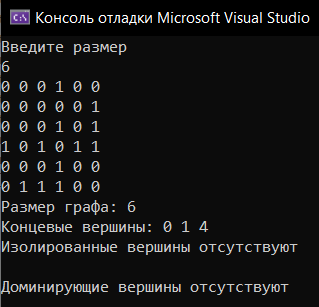


Рис.2 - Результат работы программы

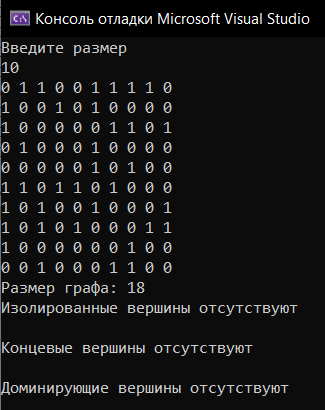


Рис.2 - Результат работы программы

Вывод:

В ходе работы была написана программа, которая генерирует и выводит на экран неориентированный граф, определяет его размер, а также находит его изолированные, концевые и доминирующие вершины.