МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра информационных технологий**

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 11**

**по дисциплине  
 «АЛГОРИТМЫ И АНАЛИЗ СЛОЖНОСТИ»**

Выполнил студент группы 24/1                                       А.А.Иванов

Направление подготовки  02.03.03  Математическое обеспечение и администрирование информационных систем

Курс    2

Отчет принял доктор физико-математических наук, профессор                                                                                       А.И. Миков

Краснодар

2022 г.

**Задание:** написать функцию, которая по заданной матрице смежности ориентированного графа определяет, является ли граф корневым бинарным деревом.

**Решение:** напишем программу, в которой ключевую роль сыграет вектор. В него мы будем записывать вершину, из которой выходит ребро, и вершину, в которую оно заходит (образуем пары). Далее просто обрабатываем полученные пары, и получаем ответ.

**Код программы:**

#include <iostream>

#include <string>

#include <algorithm>

#include <vector>

#include <random>

#include <cstdlib>

#include <ctime>

using namespace std;

struct Graph

{

vector < pair <int, int>> sosedi;

};

int main()

{

setlocale(LC\_ALL, "rus");

srand(time(NULL));

cout << "Введите количество вершин графа:" << endl;

int n;

cin >> n;

int\*\* A = new int\* [n];

for (int i = 0; i < n; i++)

A[i] = new int[n];

cout << "Введите матрицу смежности графа:" << endl;

for (int i = 0; i < n; i++)

for (int j = 0; j < n; j++)

cin >> A[i][j];

Graph graph;

for (int i = 0; i < n; i++)

for (int j = 0; j < n; j++)

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

graph.sosedi.push\_back(make\_pair(i, j));

bool f = true;

for (int i = 0; i < graph.sosedi.size(); i++)

{

if (graph.sosedi[i].first == graph.sosedi[i].second)

{

f = false;

break;

}

int otkyda = graph.sosedi[i].first;

int kol = 0;

for (int j = 0; j < graph.sosedi.size(); j++)

{

if (otkyda == graph.sosedi[j].first)

kol++;

if (kol >= 3)

{

f = false;

break;

}

if (i != j)

{

if (graph.sosedi[i].second == graph.sosedi[j].second)

{

f = false;

break;

}

if (graph.sosedi[i].first == graph.sosedi[j].second && graph.sosedi[i].second == graph.sosedi[j].first)

{

f = false;

break;

}

}

}

if (f == false)

break;

}

cout << "Рёбра вашего графа:" << endl;

for (int i = 0; i < graph.sosedi.size(); i++)

cout << "Откуда: " << graph.sosedi[i].first << " Куда: " << graph.sosedi[i].second << endl;

if (f == true)

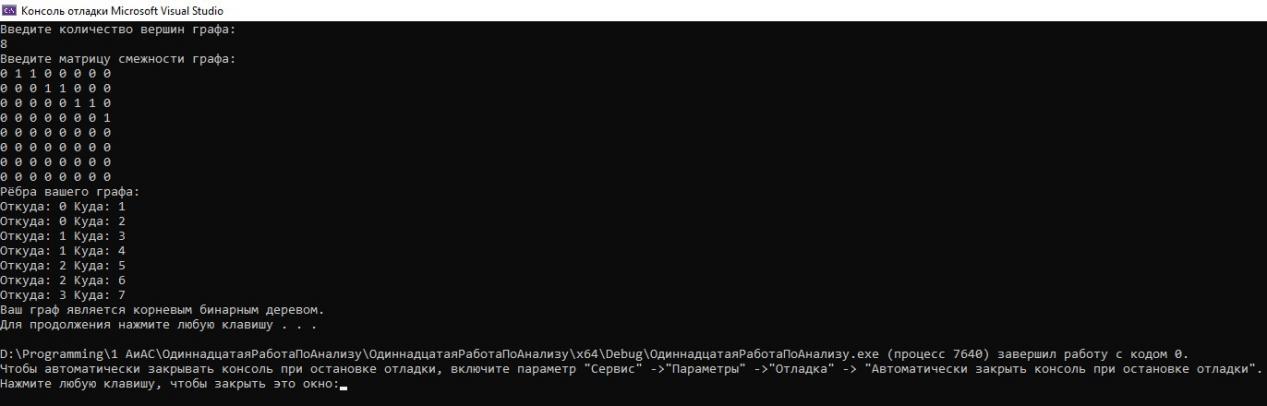
cout << "Ваш граф является корневым бинарным деревом." << endl;

else cout << "Ваш граф не является корневым бинарным деревом." << endl;

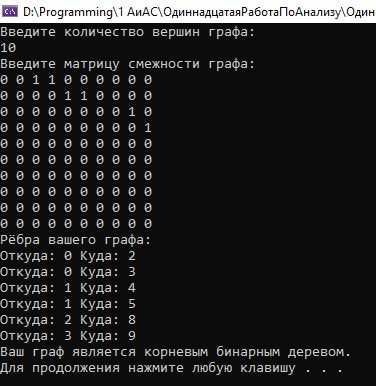
system("pause");

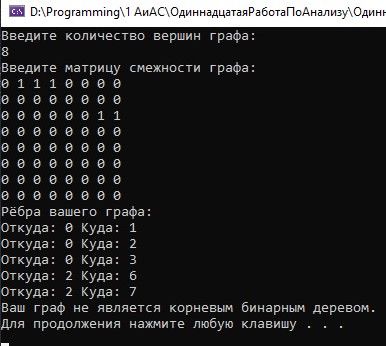
}

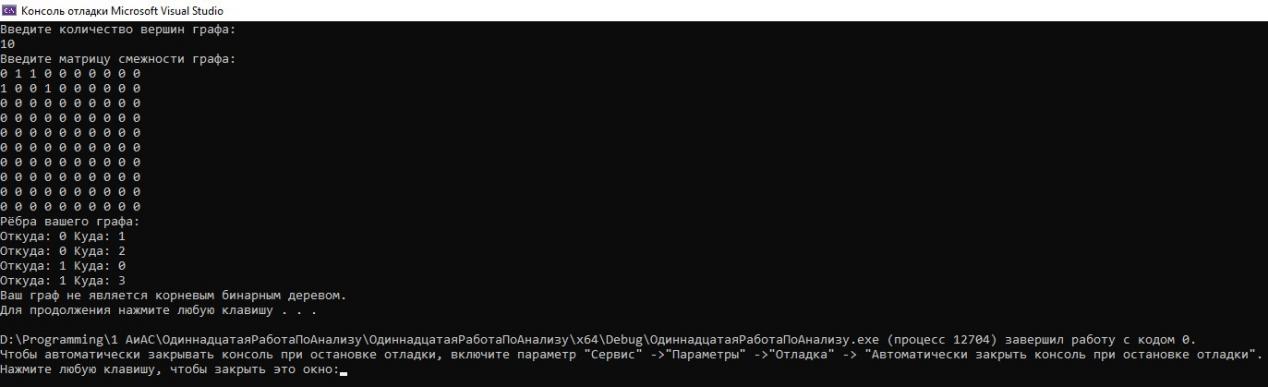
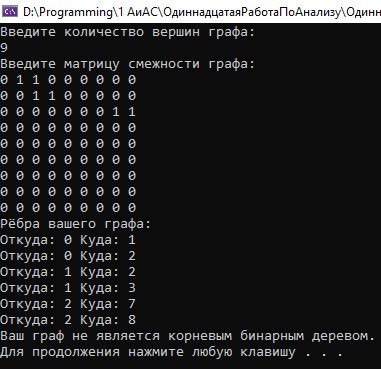
**Примеры вывода в консоль:**











**Вывод:** мы получили алгоритм, который корректно (вроде бы) распознаёт в графах корневое бинарное дерево.