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 № 2**

**по дисциплине  
 «Параллельное и низкоуровневое программирование»**

Выполнил студент группы 25/2                                       А.А. Козин

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 1**: Изоморфизм графов.

**Решение**: создадим случайный граф. Потом сделаем отображение вершин, то есть какая вершина во втором графе будет соответствовать вершине в первом и получается, что графы всегда будут изоморфны.

**Текст программы:**

#include <iostream>

#include <vector>

#include <ctime>

#include <thread>

#include <chrono>

#include <algorithm>

using namespace std;

using namespace std::chrono;

int sam(vector<vector<int>> graph, int j) {

int sum = 0;

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

sum += graph[j][i];

return sum;

}

bool isomorphism(vector<vector<int>> graphf, vector<vector<int>> graphs, int start, int end){

int k = 0;

for (int i = start; i < end; i++) {

vector<int> vfirst;

for (int j = 0; j < graphf.size(); j++) {

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

vfirst.push\_back(sam(graphf, j));

}

sort(vfirst.begin(), vfirst.end());

if (k < i + 1) {

for (int c = 0; c < graphs.size(); c++) {

vector<int> vsecond;

for (int d = 0; d < graphs.size(); d++) {

if (graphf[i][d] == 1)

vsecond.push\_back(sam(graphs, d));

}

sort(vsecond.begin(), vsecond.end());

if (vfirst == vsecond)

k++;

}

}

}

if (graphf.size() == k)

return true;

else

return false;

}

void main() {

srand(time(0));

int n;cin >> n; cout << endl;

vector<vector<int>> graphf(n, vector<int>(n));

vector<vector<int>> graphs(n, vector<int>(n));

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

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

if (i == j) {

graphf[i][j] = 0;

graphs[i][j] = 0;

}

else {

graphf[i][j] = 1; graphf[j][i] = 1;

graphs[i][j] = 1; graphs[j][i] = 1;

}

}

}

int r1 = rand() % n; int r2 = rand() % n;

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

{

graphf[r1][i] = 0; graphf[i][r1] = 0;

graphs[r2][i] = 0; graphs[i][r2] = 0;

}

int r3 = rand() % n; int r4 = rand() % n;

graphf[r1][r3] = 1; graphf[r3][r1] = 1;

graphs[r2][r4] = 1; graphs[r4][r2] = 1;

auto t0 = high\_resolution\_clock::now();

bool ans = isomorphism(graphf, graphs, 0, n);//true,false isomorph

auto t1 = high\_resolution\_clock::now();

cout << "isomorphism is: " << ans << endl;

cout << "duration\_cast vertexes: " << duration\_cast<milliseconds>(t1 - t0).count() << " msec" << endl;

thread first(isomorphism, graphf, graphs, 0, n / 2);

thread second(isomorphism, graphf, graphs, n / 2, n);

t0 = high\_resolution\_clock::now();

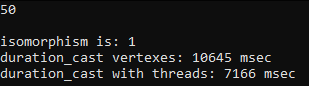
first.join(); second.join();

t1 = high\_resolution\_clock::now();

cout << "duration\_cast with threads: " << duration\_cast<milliseconds>(t1 - t0).count() << " msec" << endl;

}

**Программа выдает в консоли результаты в следующем виде:**



Ответ выводиться в виде булевой переменной, где 1 – граф изоморфен, а 0 – нет.