Министерство образования Республики Беларусь

Учреждение образования

«Брестский Государственный технический университет»

Кафедра ИИТ

**Лабораторная работа №2**

По дисциплине «Математические основы интеллектуальных систем»

Тема: «Нахождение эйлеровых и гамильтоновых циклов в неориентированном графе»

**Выполнил:**

Студент 2 курса

Группы ИИ-21

Литвинюк Т. В.

**Проверил:**

Козинский А. А.

Брест 2022

**Цель:** научиться находить элеровые и гамильтоновые циклы в графе.

**Ход работы:**

**Вариант 7**

#include <iostream>

using namespace std;

class stack{

public:

int length=0;

int stack[100];

int get(){

return stack[length-- - 1];

}

void add(int num){

stack[length++] = num;

}

int check(){

return stack[length-1];

}

};

class queue{

public:

int q[100];

int length = 0, left = 0, right = 0;

void add(int num){

q[right++] = num; length++;

}

int get(){

length--;

return q[left++];

}

void swap(int pos1, int pos2){

int t = q[pos1];

q[pos1] = q[pos2];

q[pos2] = t;

}

};

class list{

public:

int list[100];

int length=0;

void append(int num){

list[length++] = num;

}

int last(){

return list[length-1];

}

};

bool isVertexHasNeighbors(int (\*graph)[6], int vertex){

for (int i = 0; i < 6; i++){

if (graph[vertex][i] == 1)

return true;

}

return false;

}

// ############################################################

void findEulerianCycle(int (\*matrix)[6], int vertex){

int graph[6][6];

for (int i = 0; i < 6; i++){

for (int j = 0; j < 6; j++){

graph[i][j] = matrix[i][j];

}

}

stack S;

list C;

S.add(vertex);

while (true){

if (!isVertexHasNeighbors(graph, S.check()))

{C.append(S.get()); vertex = S.check();}

else {

for (int i = 0; i < 6; i++){

if (graph[vertex][i] == 1){

graph[vertex][i] = 0; graph[i][vertex] = 0;

S.add(i); vertex = i; break;

}

}

}

if (C.list[0] == C.last() and C.length>1){

for (int i = 0; i < C.length; i++)

cout << C.list[i]+1;

cout << endl;

break;

}

}

}

// ############################################################

void findGamiltonCycle(int (\*matrix)[7], int vertex){

queue Q;

list C;

for (int i = 0; i < 7; i++){

Q.add(i);

}

int i;

for (int j = 0; j < 7 \* 6; j++){

if (matrix[Q.q[0]][Q.q[1]] != 1){

i = 1;

while (matrix[Q.q[0]][Q.q[i]] != 1 || matrix[Q.q[1]][Q.q[i+1]] != 1){

i++;

}

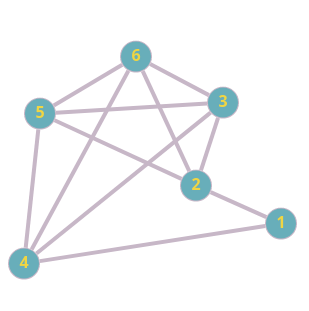
for (int k = 0; 1 + k < i - k; k++){

Q.swap(1 + k, i - k);

}

Q.add(Q.get());

}

}

for (i = 0; i < 7; i++)

cout << Q.q[i];

cout << endl;

}

// ############################################################

int main() {

const int tops1 = 6, edges1 = 11;

int adjacencyMatrix1[tops1][tops1] = {{0, 1, 0, 1, 0, 0},

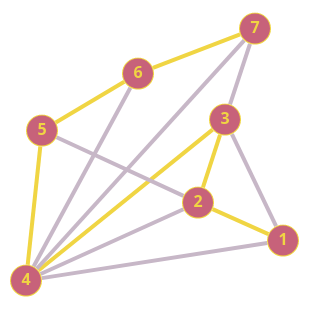
{1, 0, 1, 0, 1, 1},

{0, 1, 0, 1, 1, 1},

{1, 0, 1, 0, 1, 1},

{0, 1, 1, 1, 0, 1},

{0, 1, 1, 1, 1, 0}};

findEulerianCycle(adjacencyMatrix1, 0);

const int tops2 = 7, edges2 = 13;

int adjacencyMatrix2[tops2][tops2] = {{0, 1, 1, 1, 0, 0, 0},

{1, 0, 1, 1, 1, 0, 0},

{1, 1, 0, 1, 0, 0, 1},

{1, 1, 1, 0, 1, 1, 1},

{0, 1, 0, 1, 0, 1, 0},

{0, 0, 0, 1, 1, 0, 1},

{0, 0, 1, 1, 0, 1, 0}};

findGamiltonCycle(adjacencyMatrix2, 0);

}

**Вывод:** в ходе лабораторной работы я научился находить эйлеров и гамильтонов цикл в графе.