**Лабораторная работа 1 по Теории графов**

Студент Маслова Анастасия Сергеевна

Группа НКНбд-01-21

Преподаватели: Доцент Маркова Е.В.

Тема лабораторной работы: Алгоритм поиска Эйлерова цикла в графе

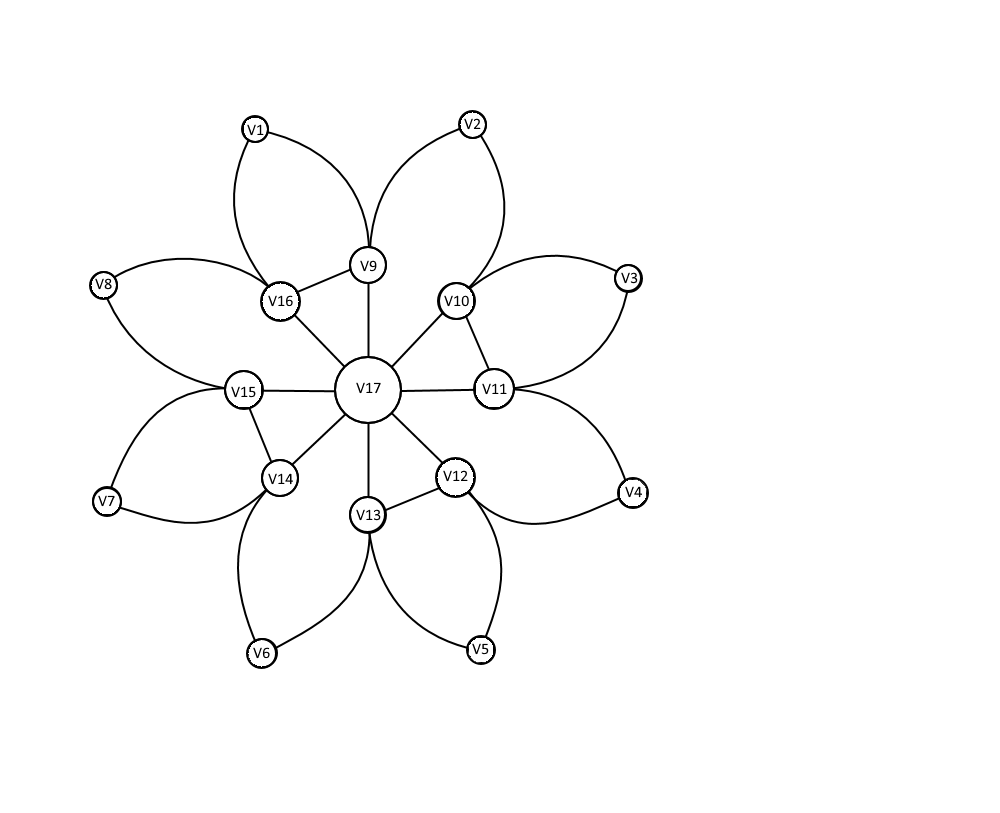
Количество баллов: \_\_\_\_\_\_ баллов из 10 б.

1. **Теоретические сведения по алгоритму поиска Эйлерова цикла в графе**

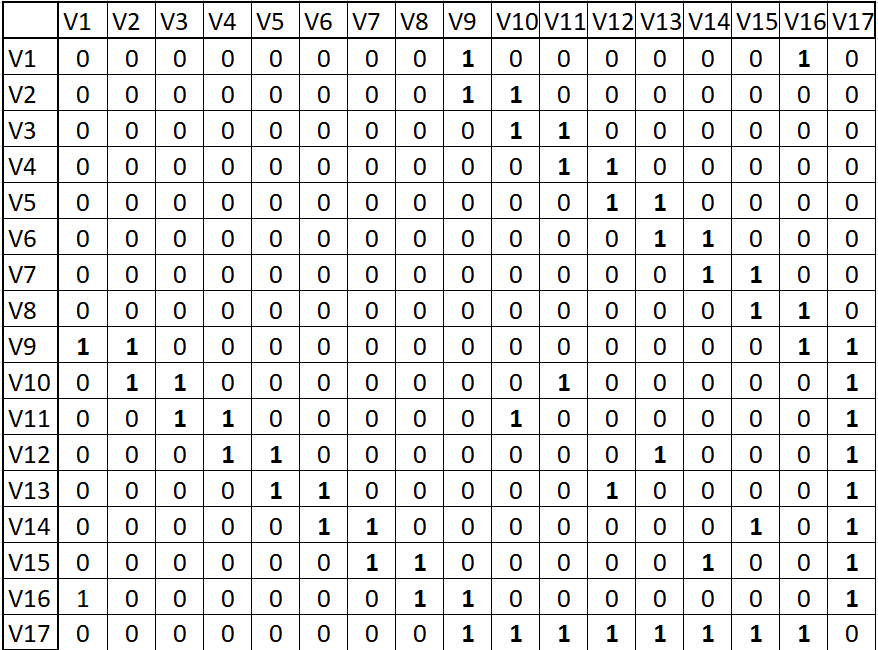
Для лабораторной работы использовались источники [1,2].

1. **Реализация алгоритма поиска Эйлерова цикла в графе на примере**

Для реализации алгоритма поиска Эйлерова цикла в графе был выбран граф на рис. 1, для которого была найдена матрица смежности (рис.2).



**рис. 1 Изначальный граф**



**рис. 2 Матрица смежности**

Для данного графа применен представленный ниже исходный код алгоритма поиска Эйлерова цикла в графе, выполненный на языке программирования С++.

Исходный код:

using namespace std;

#include <iostream>

struct Node

{

int inf;

Node\* next;

};

void push(Node\*& st, int dat)

{

Node\* el = new Node;

el->inf = dat;

el->next = st;

st = el;

}

int pop(Node\*& st)

{

int value = st->inf;

Node\* temp = st;

st = st->next;

delete temp;

return value;

}

int peek(Node\* st)

{

return st->inf;

}

Node\*\* graph;

const int vertex = 1;

void add(Node\*& list, int data)

{

if (!list) { list = new Node; list->inf = data; list->next = 0; return; }

Node\* temp = list;

while (temp->next)temp = temp->next;

Node\* elem = new Node;

elem->inf = data;

elem->next = NULL;

temp->next = elem;

}

void del(Node\*& l, int key)

{

if (l->inf == key) { Node\* tmp = l; l = l->next; delete tmp; }

else

{

Node\* tmp = l;

while (tmp)

{

if (tmp->next)

if (tmp->next->inf == key)

{

Node\* tmp2 = tmp->next;

tmp->next = tmp->next->next;

delete tmp2;

}

tmp = tmp->next;

}

}

}

int eiler(Node\*\* gr, int num)

{

int count;

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

{

count = 0;

Node\* tmp = gr[i];

while (tmp)

{

count++;

tmp = tmp->next;

}

if (count % 2 == 1)return 0;

}

return 1;

}

void eiler\_path(Node\*\* gr)

{

Node\* S = NULL;

int v = vertex;

int u;

push(S, v);

while (S)

{

v = peek(S);

if (!gr[v])

{

v = pop(S); cout << v + 1 << " ";

}

else

{

u = gr[v]->inf; push(S, u);

del(gr[v], u); del(gr[u], v);

}

}

}

int main()

{

int i;

system("CLS");

cout << "The number of vertices of the graph : "; int n; cin >> n;

int zn;

cout << "Enter the adjacency matrix: " << endl;

graph = new Node \* [n];

for (int i = 0; i < n; i++)graph[i] = NULL;

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

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

{

cin >> zn;

if (zn) add(graph[i], j);

}

cout << "\nThe result: ";

if (eiler(graph, n))eiler\_path(graph);

else cout << "The graph is not Eulerian.";

cout << endl;

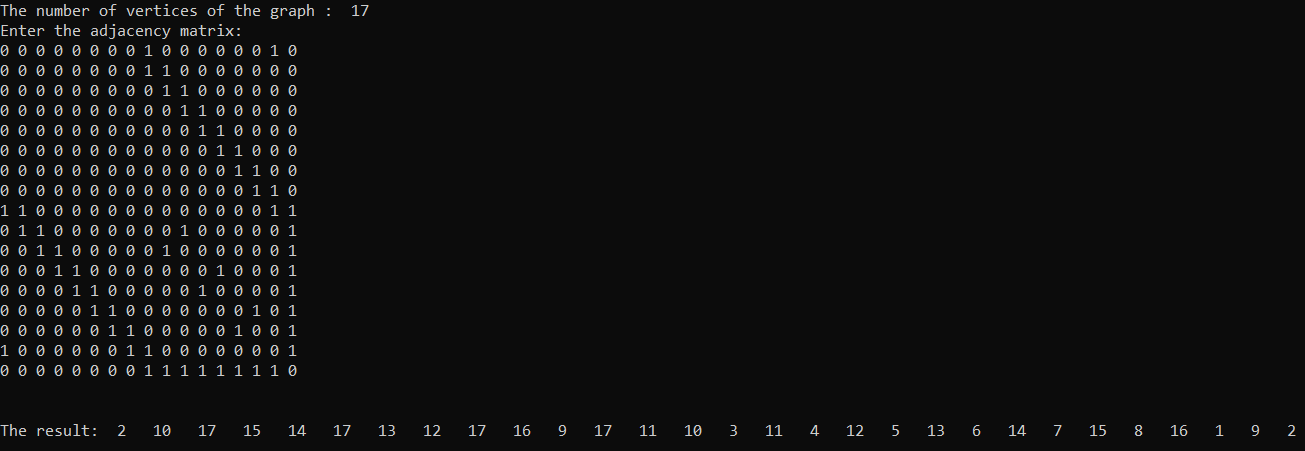
cin.get();

cin.get();

return(0);

}

Итог работы по нахождению Эйлерова цикла (рис.3):



**рис. 3 Результат работы программы**

**Источники**

1) Зарипова Э.Р., Кокотчикова М.Г. Лекции по дискретной математике: Теория графов. Учебное пособие. М., изд-во: РУДН, 2013, 162 с.

2)Учебный портал РУДН, раздел «Теория конечных графов» https://esystem.rudn.ru/course/view.php?id=17753