МІНІСТЕРСТВО ОСВІТИ І НАУКИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Інститут комп'ютерних наук та інформаційних технологій

Кафедра систем штучного інтелекту



Лабораторна робота №4

з курсу “Дискретна математика ”

Виконав:  
ст. гр.  КН-110

Чорній Юрій

Викладач:

Мельникова Н.І.

Львів – 2018

**Варіант 15**

**Основні операції над графами. Знаходження остова мінімальної ваги за алгоритмом Пріма-Краскала**

**Мета:** набуття практичних вмінь та навичок з використання алгоритмів Пріма і Краскала.

**Завдання № 1**. Розв'язати на графах наступні задачі:

1. Виконати наступні операції над графами:

1) знайти доповнення до першого графу,

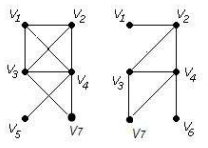
2) об’єднання графів,

3) кільцеву суму G1 та G2 (G1+G2),

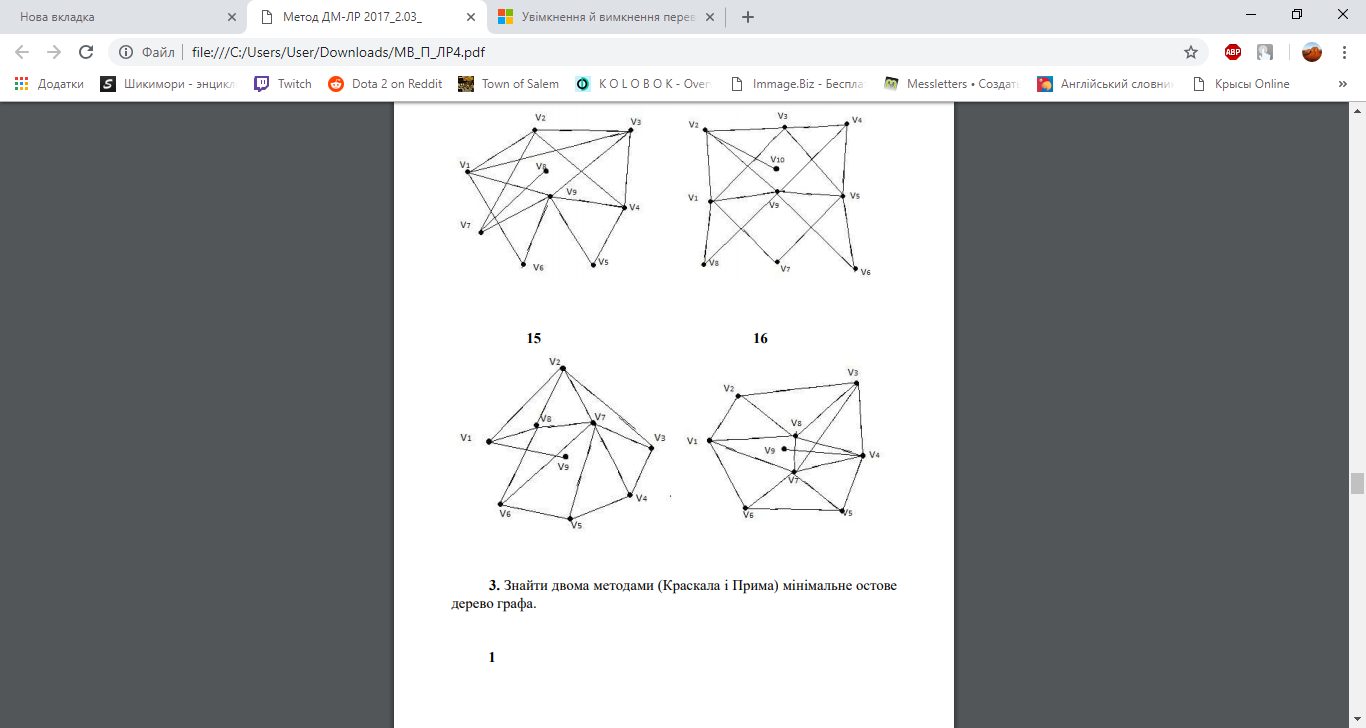
4) розщепити вершину у другому графі,

5) виділити підграф А, що складається з 3-х вершин в G1 і знайти стягнення А в G1 (G1\ A),

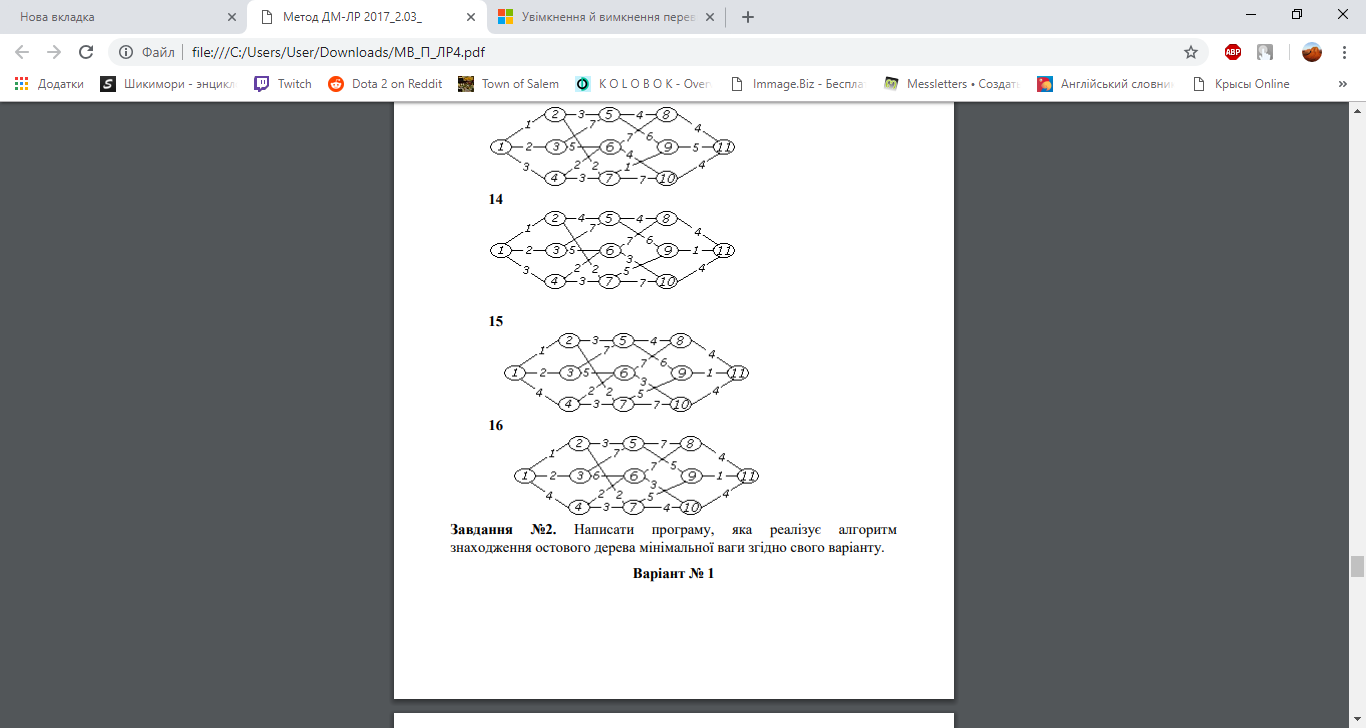
6) добуток графів



2. Знайти таблицю суміжності та діаметр графа.

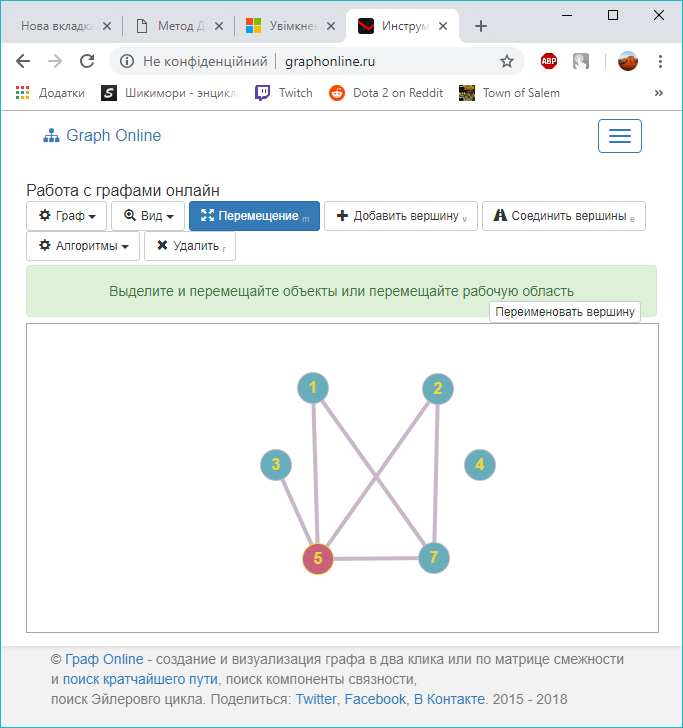


3. Знайти двома методами (Краскала і Прима) мінімальне остове дерево графа.

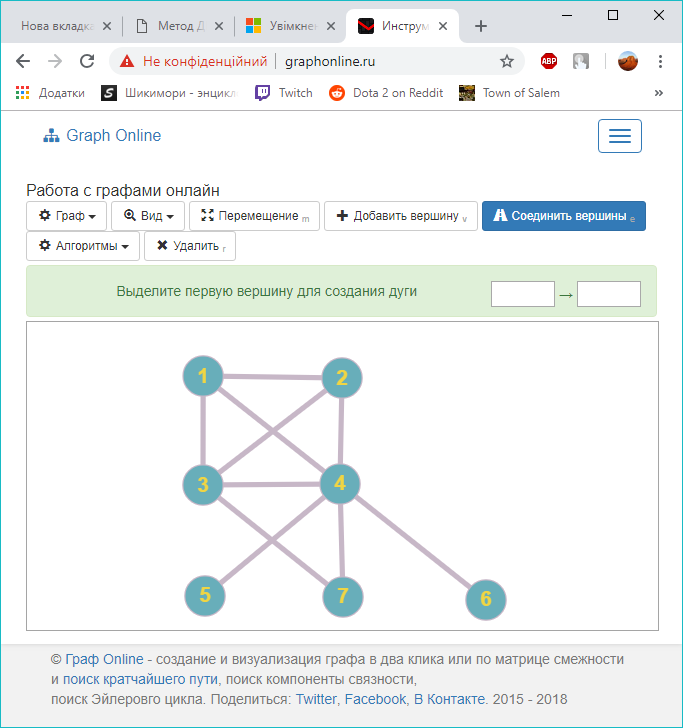


**Розв'язок:**

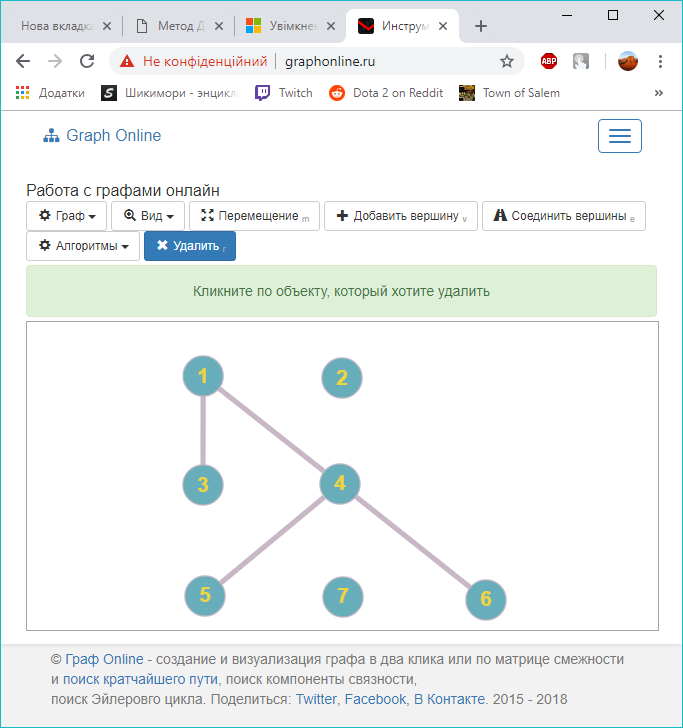
1. 1)



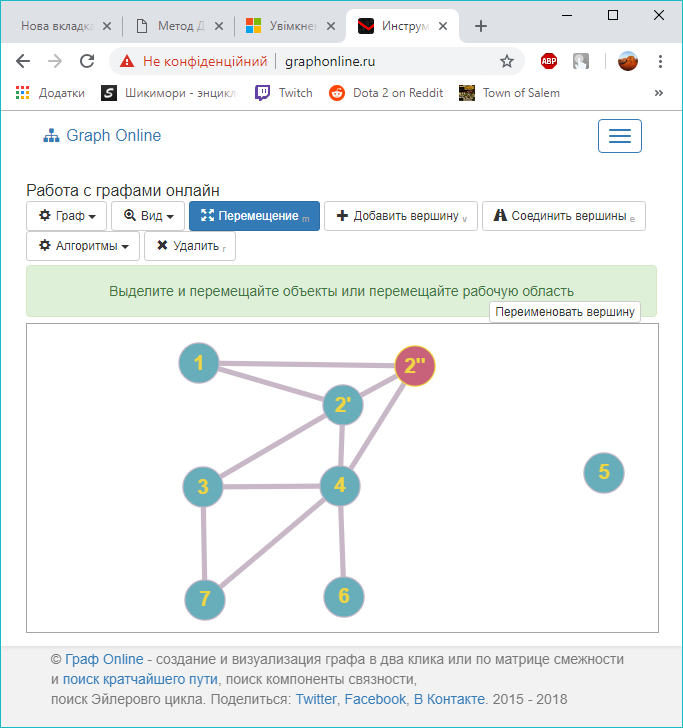
2)



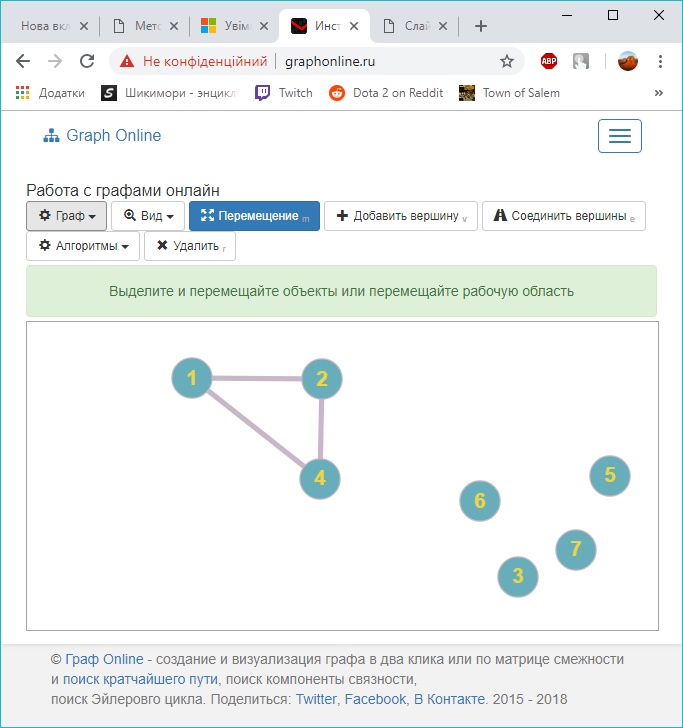
3)



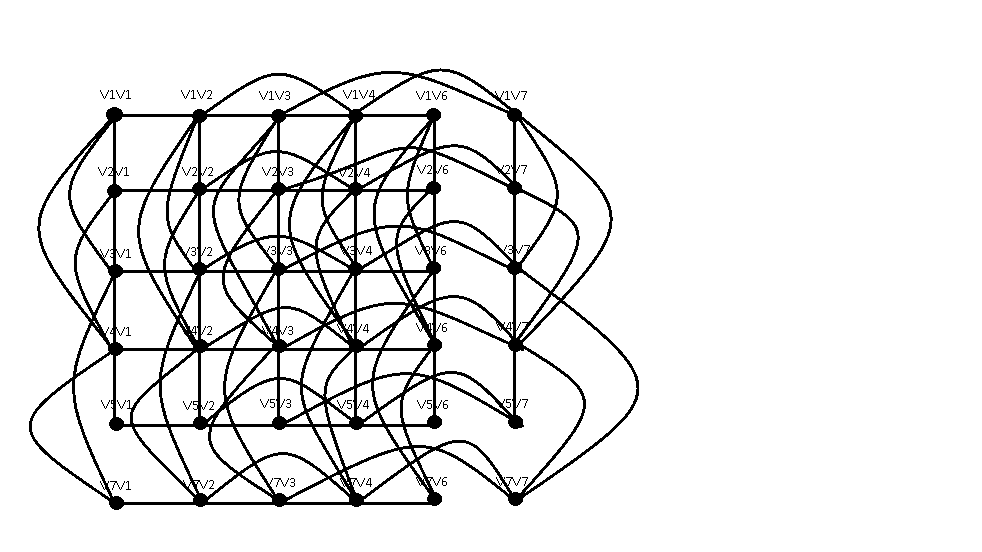
4)



5) Стягнення V3, V5, V7 до V4 y G1



6)



1. 1) Матриця суміжності

V1 V2 V3 V4 V5 V6 V7 V8 V9

V1 0, 1, 0, 0, 0, 0, 0, 1, 0,

V2 1, 0, 1, 0, 0, 0, 1, 0, 1,

V3 0, 1, 0, 1, 0, 0, 0, 0, 1,

V4 0, 0, 1, 0, 1, 0, 0, 0, 1,

V5 0, 0, 0, 1, 0, 1, 0, 0, 1,

V6 0, 0, 0, 0, 1, 0, 1, 0, 1,

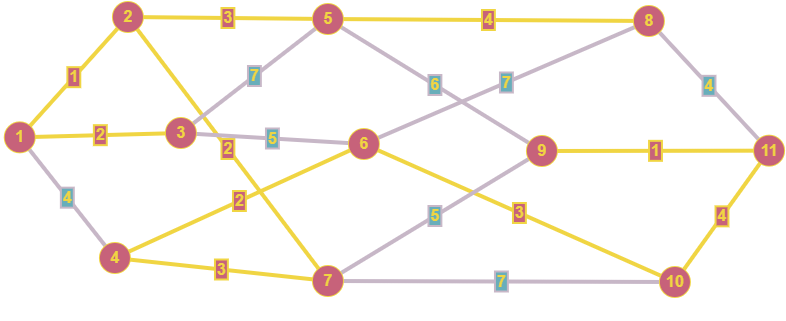
V7 0, 1, 0, 0, 0, 1, 0, 0, 1,

V8 1, 0, 0, 0, 0, 0, 0, 0, 0,

V9 0, 1, 1, 1, 1, 1, 1, 0, 0,

2)Діаметр графа d = 4 (4 -> 3 -> 2 -> 1 -> 9)

№3



#include<stdio.h>

#include<stdlib.h>

#define infinity 9999

#define MAX 20

int G[MAX][MAX],spanning[MAX][MAX],n;

int Prima()

{

int cost[MAX][MAX];

int u,v,min\_distance,distance[MAX],from[MAX];

int visited[MAX],no\_of\_edges,i,min\_cost,j;

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

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

{

if(G[i][j]==0)

cost[i][j]=infinity;

else

cost[i][j]=G[i][j];

spanning[i][j]=0;

}

distance[0]=0;

visited[0]=1;

for(i=1;i<n;i++)

{

distance[i]=cost[0][i];

from[i]=0;

visited[i]=0;

}

min\_cost=0;

no\_of\_edges=n-1;

while(no\_of\_edges>0)

{

min\_distance=infinity;

for(i=1;i<n;i++)

if(visited[i]==0&&distance[i]<min\_distance)

{

v=i;

min\_distance=distance[i];

}

u=from[v];

printf("%i-->%i \n",u,v);

spanning[u][v]=distance[v];

spanning[v][u]=distance[v];

no\_of\_edges--;

visited[v]=1;

for(i=1;i<n;i++)

if(visited[i]==0&&cost[i][v]<distance[i])

{

distance[i]=cost[i][v];

from[i]=v;

}

min\_cost=min\_cost+cost[u][v];

}

return(min\_cost);

}

int main()

{

int i,j,total\_cost;

printf("Enter the number of vertices:");

scanf("%d",&n);

printf("\nEnter the adjacency matrix:\n");

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

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

scanf("%d",&G[i][j]);

printf("\nSteps begins with 0,so 0 step its like first(1)\n");

printf("\n");

total\_cost=Prima();

printf("\nspanning tree matrix:\n");

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

{

printf("\n");

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

printf("%d\t",spanning[i][j]);

}

printf("\n\nTotal cost of spanning tree=%d",total\_cost);

printf("\n");

return 0;

}

