

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

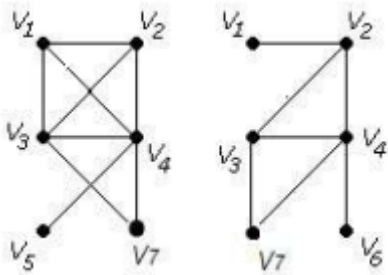
## Варіант 15

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

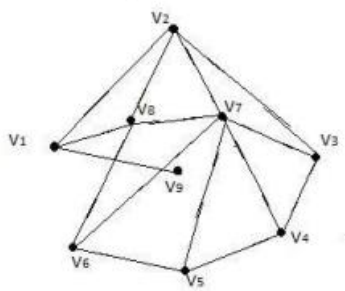
**Мета:** набуття практичних вмінь та навичок з використання алгоритмів Пріма і Краскала.

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

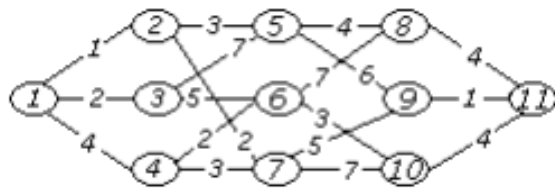
1. Виконати наступні операції над графами:
  - 1) знайти доповнення до першого графу,
  - 2) об'єднання графів,
  - 3) кільцеву суму  $G_1$  та  $G_2$  ( $G_1+G_2$ ),
  - 4) розщепити вершину у другому графі,
  - 5) виділити підграф  $A$ , що складається з 3-х вершин в  $G_1$  і знайти стягнення  $A$  в  $G_1$  ( $G_1 \setminus A$ ),
  - 6) добуток графів



2. Знайти таблицю суміжності та діаметр графа.

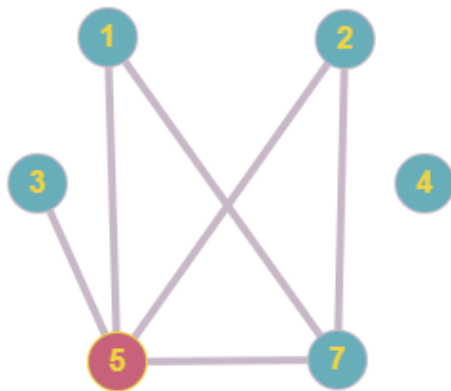


3. Знайти двома методами (Краскала і Прима) мінімальне остове дерево графа.

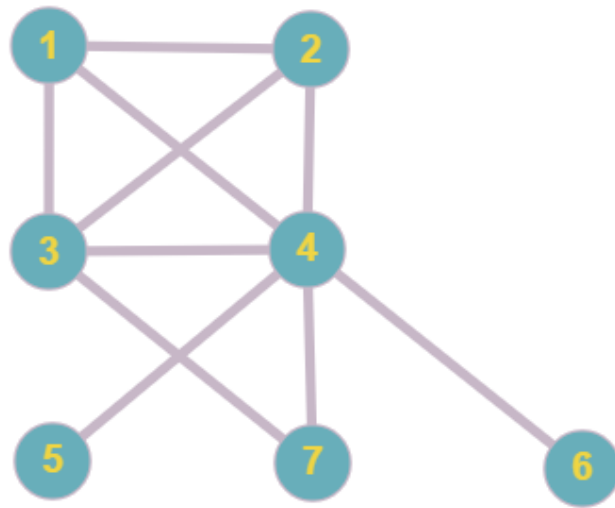


**Розв'язок:**

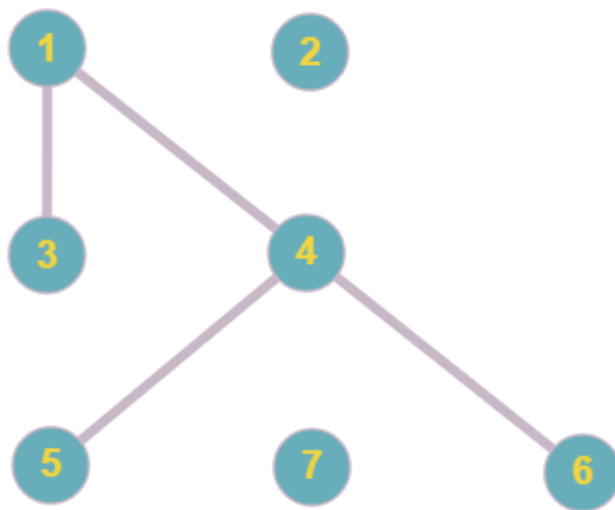
1. 1)



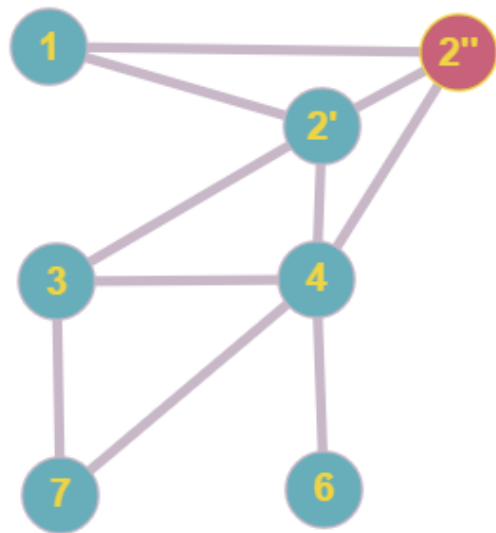
2)



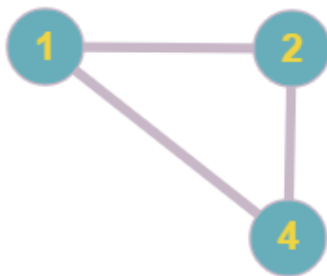
3)



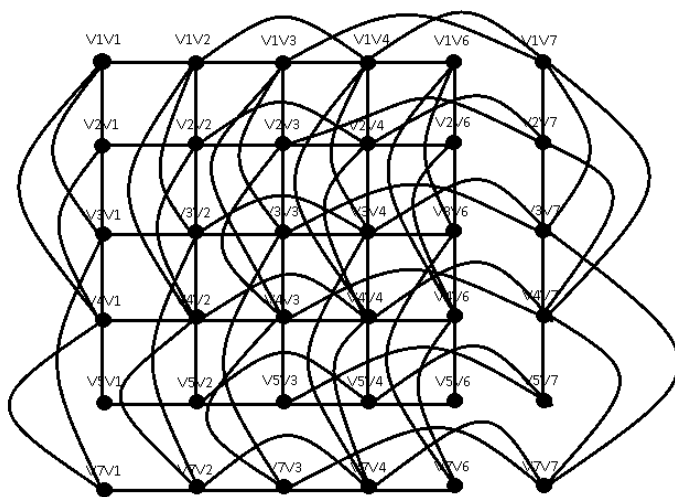
4)



5) Стягнення  $V_3, V_5, V_7$  до  $V_4$  у  $G_1$



6)



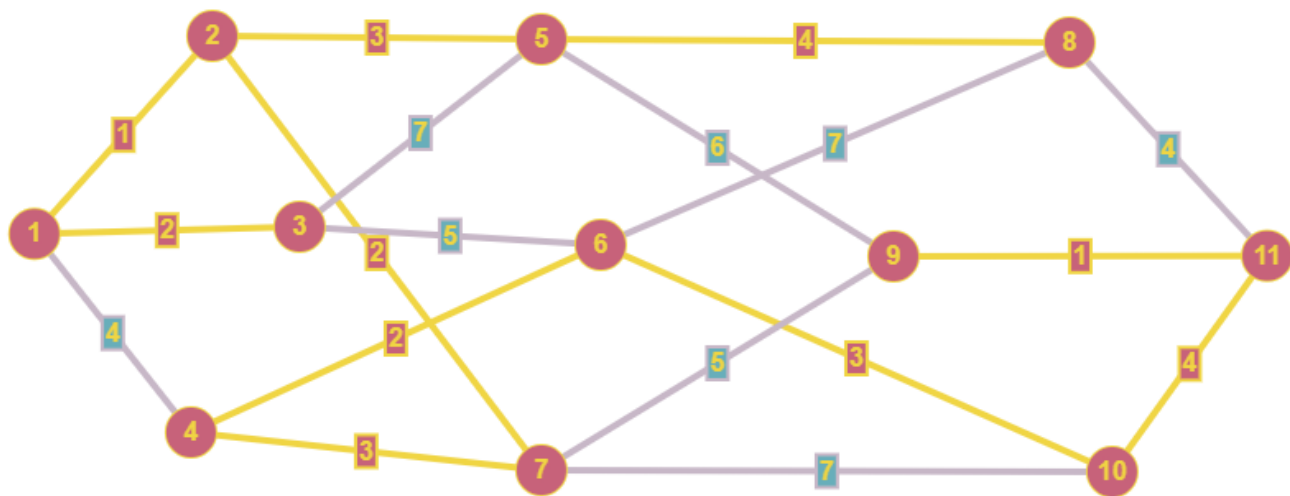
2. 1) Матриця суміжності

$v_1 \ v_2 \ v_3 \ v_4 \ v_5 \ v_6 \ v_7 \ v_8 \ v_9$   
 $v_1 \ 0, \ 1, \ 0, \ 0, \ 0, \ 0, \ 0, \ 1, \ 0,$

$v_2$  1, 0, 1, 0, 0, 0, 1, 0, 1,  
 $v_3$  0, 1, 0, 1, 0, 0, 0, 0, 1,  
 $v_4$  0, 0, 1, 0, 1, 0, 0, 0, 1,  
 $v_5$  0, 0, 0, 1, 0, 1, 0, 0, 1,  
 $v_6$  0, 0, 0, 0, 1, 0, 1, 0, 1,  
 $v_7$  0, 1, 0, 0, 0, 1, 0, 0, 1,  
 $v_8$  1, 0, 0, 0, 0, 0, 0, 0, 0,  
 $v_9$  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;
}

```



```
jharvard@appliance (~/.labalgo): ./prim
Enter the number of vertices:11
```

```
Enter the adjacency matrix:
```

```
0 1 2 3 0 0 0 0 0 0 0
1 0 0 0 4 0 2 0 0 0 0
2 0 0 0 7 5 0 0 0 0 0
3 0 0 0 0 2 3 0 0 0 0
0 4 7 0 0 0 0 4 6 0 0
0 0 5 2 0 0 0 7 0 3 0
0 2 0 3 0 0 0 0 5 7 0
0 0 0 0 4 7 0 0 0 0 4
0 0 0 0 6 0 5 0 0 0 1
0 0 0 0 0 3 7 0 0 0 4
0 0 0 0 0 0 0 4 1 4 0
```

```
Steps begins with 0,so 0 step its like first(1)
```

```
0-->1
0-->2
1-->6
0-->3
3-->5
5-->9
1-->4
4-->7
9-->10
10-->8
```