МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

ІНСТИТУТ КОМП'ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА СИСТЕМ ШТУЧНОГО ІНТЕЛЕКТУ

Лабораторна робота №4

на тему: " *Основні операції над графами. Знаходження остова мінімальної ваги за алгоритмом Пріма-Краскала* "

Виконала:

студентка групи КН-109

Чабан Софія

Прийняла:

Мельникова Н.І.

ЛЬВІВ 2018

Лабораторна робота №4

Тема роботи: Основні операції над графами. Знаходження остова мінімальної ваги за алгоритмом Пріма-Краскала

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

Варіант №12

Постановка завдання №1:

Розв'язати на графах наступні задачі: 1. Виконати наступні операції над графами:

1) знайти доповнення до першого графу,

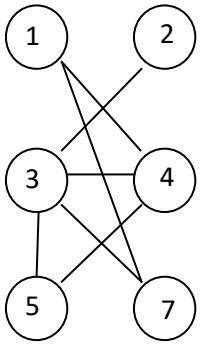
2) об’єднання графів,

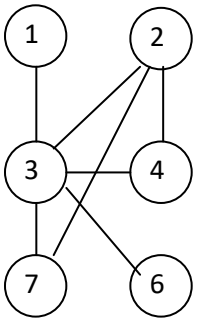
3) кільцеву суму G1 та G2 (G1+G2),

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),

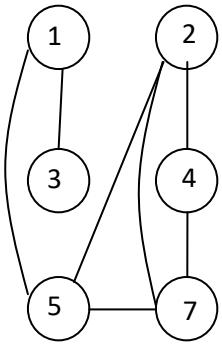
6) добуток графів.



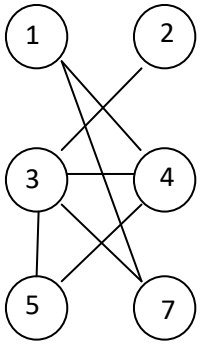


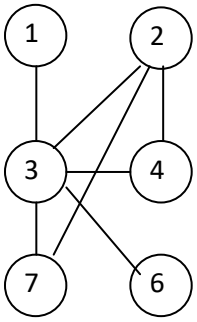
Розв’язання:

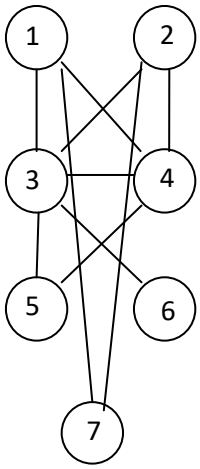
1) Доповнення до першого графу



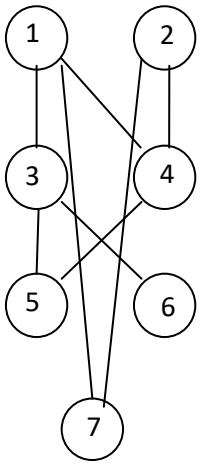
2) Об’єдання графів



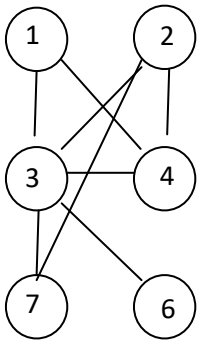


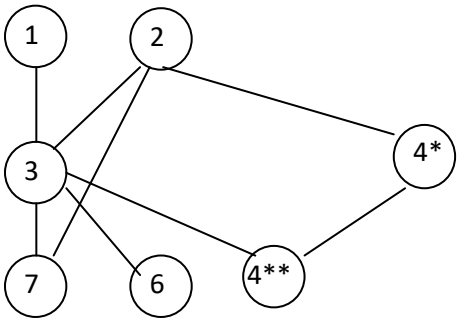


3) Кільцева сума



4)Розщепити вершину у другому графі

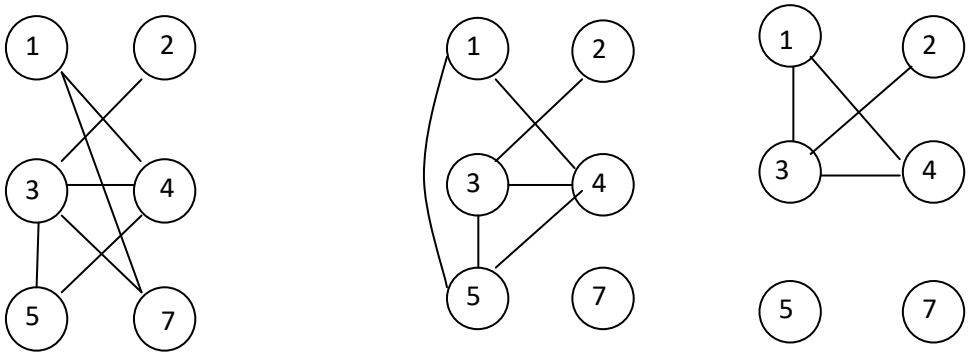




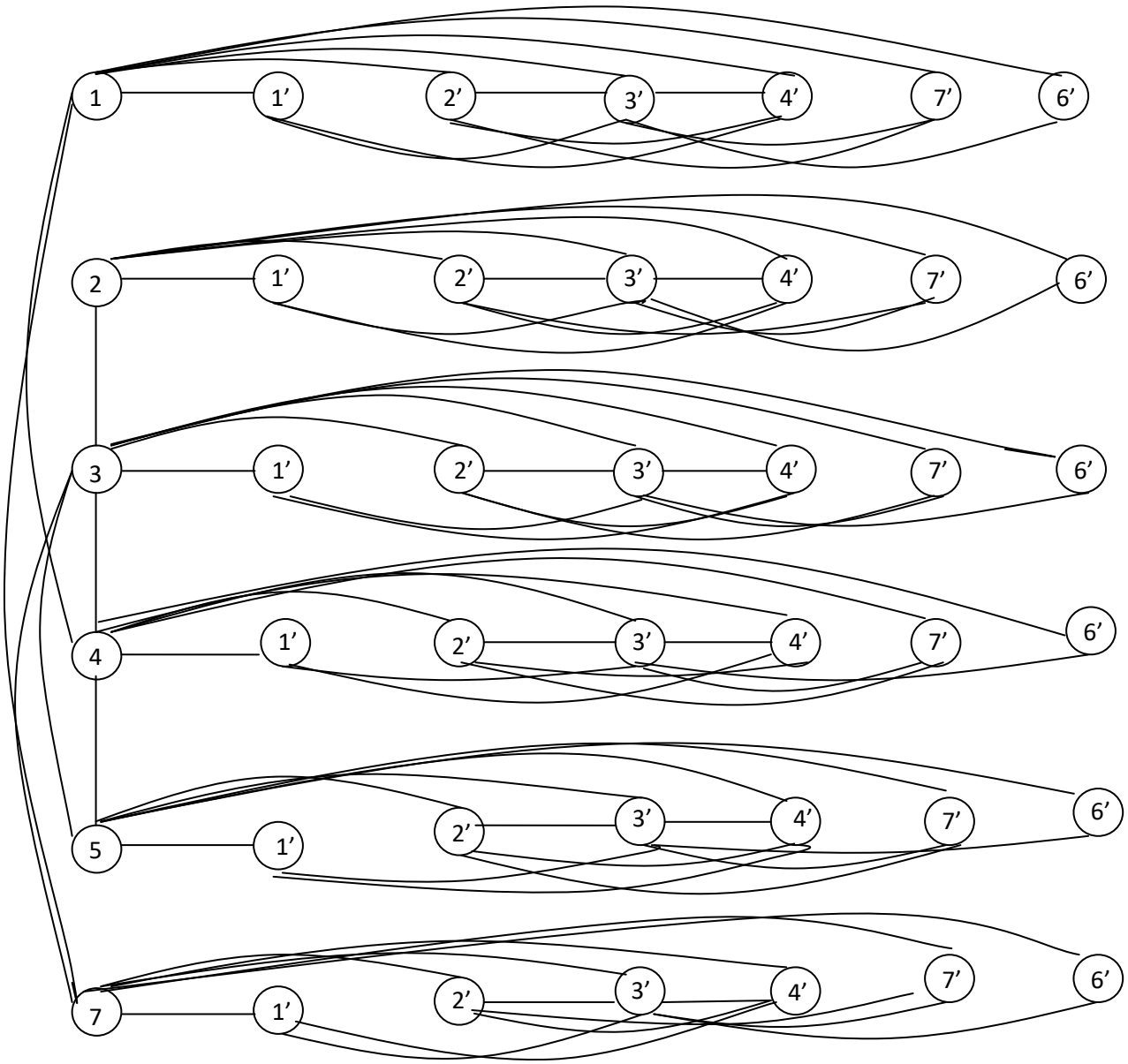
5) Виділити підграф A, що складається з 3-х вершин в G1 і знайти стягнення A в G1 (G1\ A)

Виділимо три вершини : 3, 5, 7.

Спочатку стягнемо 7 в 5, потім 5 в 3.

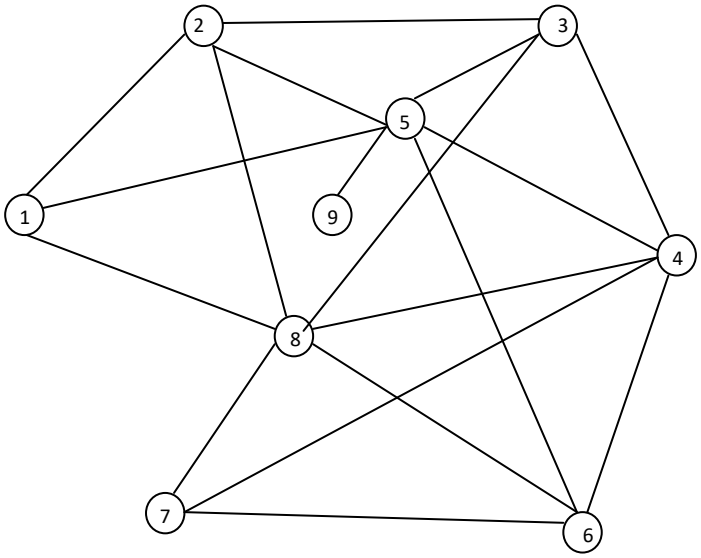


6) Добуток графів



Постановка завдання №2:

Знайти таблицю суміжності та діаметр графа.



Розв’язання:

Таблиця суміжності:

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
1	0	1	0	0	1	0	0	1	0
2	1	0	1	0	1	0	0	1	0
3	0	1	0	1	1	0	0	1	0
4	0	0	1	0	1	1	1	1	0
5	1	1	1	1	0	1	0	0	1
6	0	0	0	1	1	0	1	1	0
7	0	0	0	1	0	1	0	1	0
8	1	1	1	1	0	1	1	0	0
9	0	0	0	0	1	0	0	0	0

Для того, щоб знайти діаметр графа, побудуємо таблицю найкоротших відстаней між вершинами:

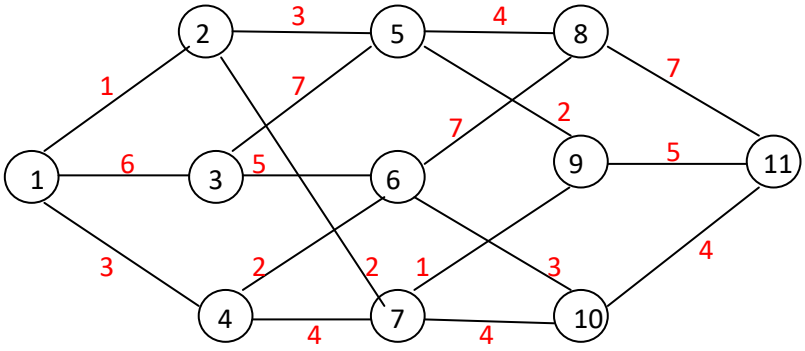
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
1	0	1	2	2	1	2	2	1	2
2	1	0	1	2	1	2	2	1	2
3	2	1	0	1	1	2	2	1	2
4	2	2	1	0	1	1	1	1	2
5	1	1	1	1	0	1	2	2	1
6	2	2	2	1	1	0	1	1	2
7	2	2	2	2	2	1	0	1	3
8	1	1	1	1	2	1	1	0	3

9	2	2	2	2	1	2	3	3	0

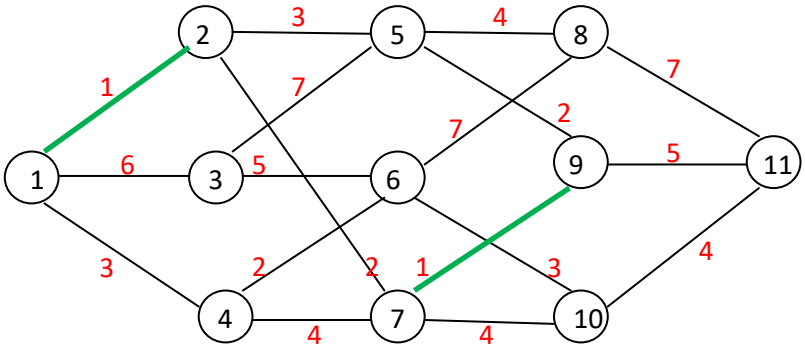
Отже, діаметр (тобто максимальна довжина найкоротшого шляху) дорівнює **3.**

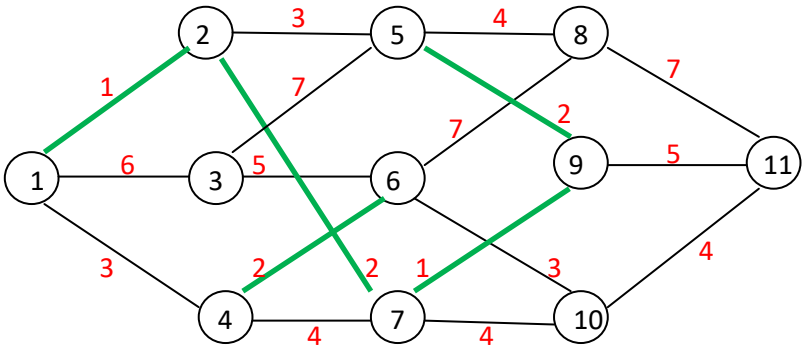
Постановка завдання №3

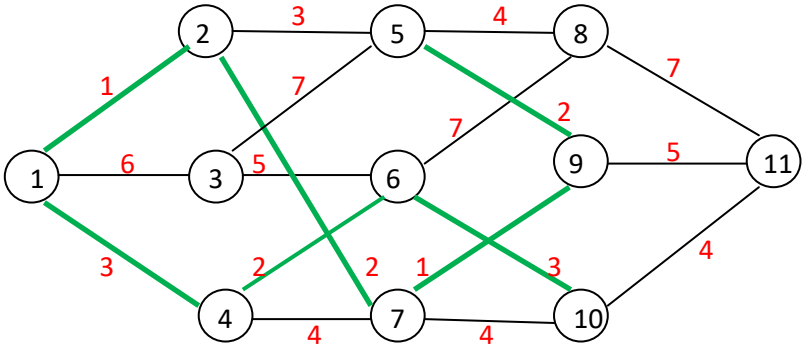
Знайти двома методами (Краскала і Прима) мінімальне остове дерево графа.

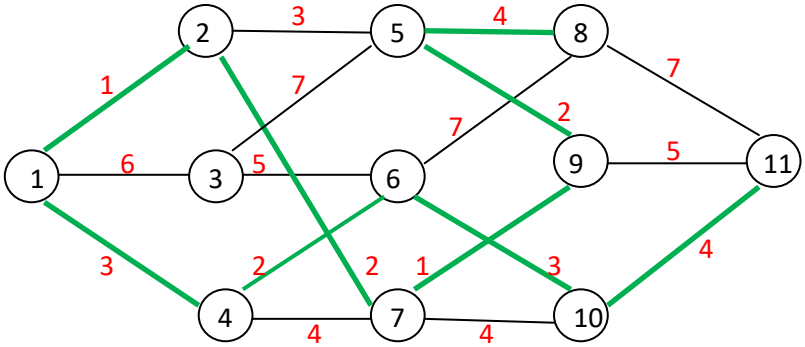


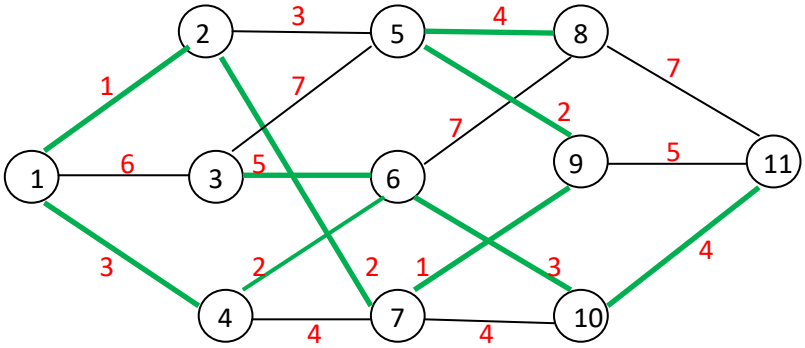
1)Методом Краскала

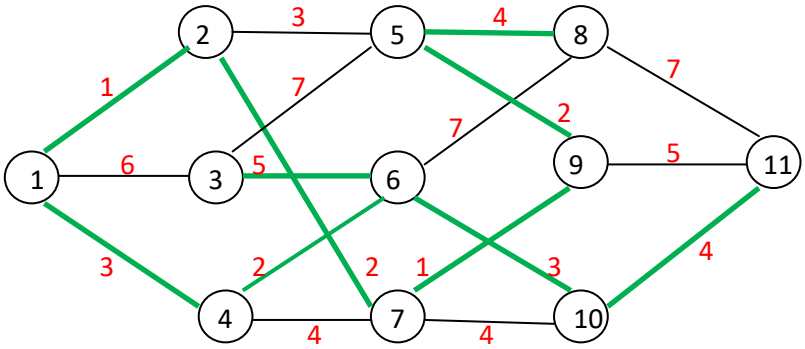


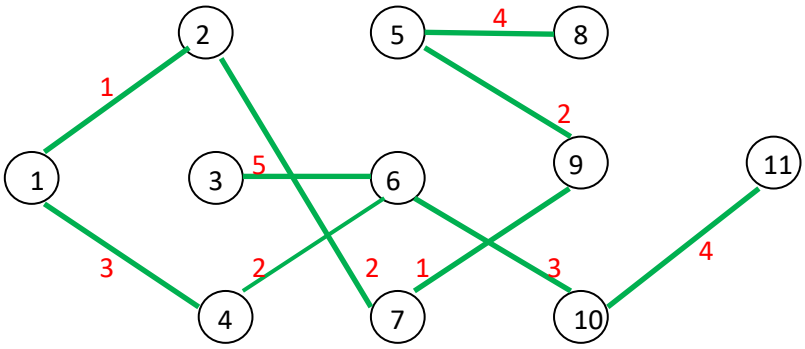






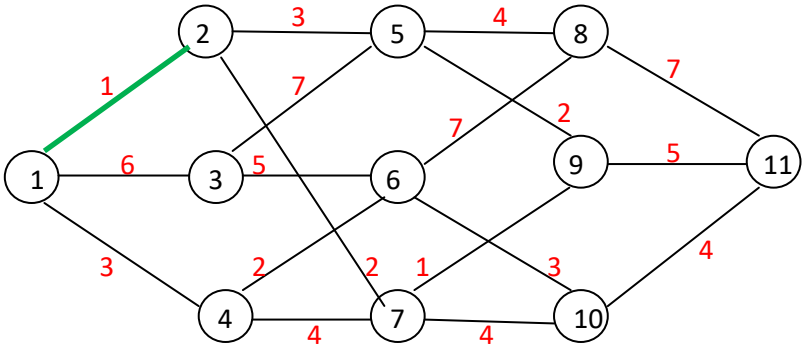


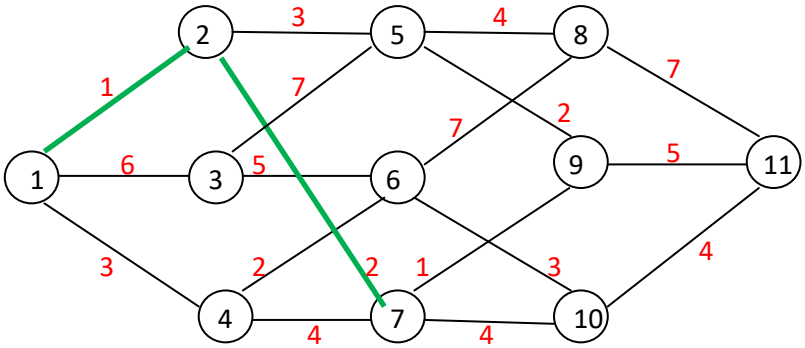


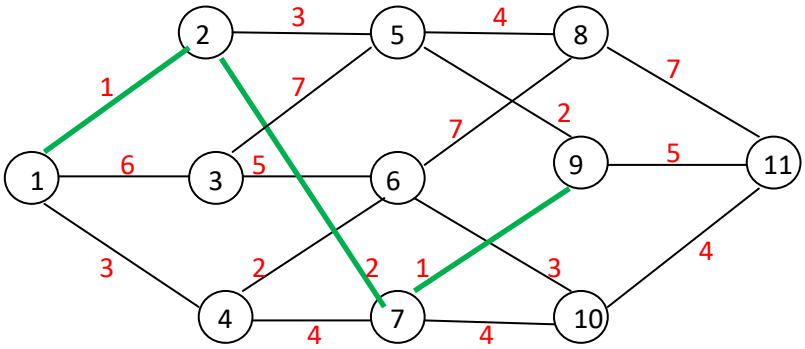


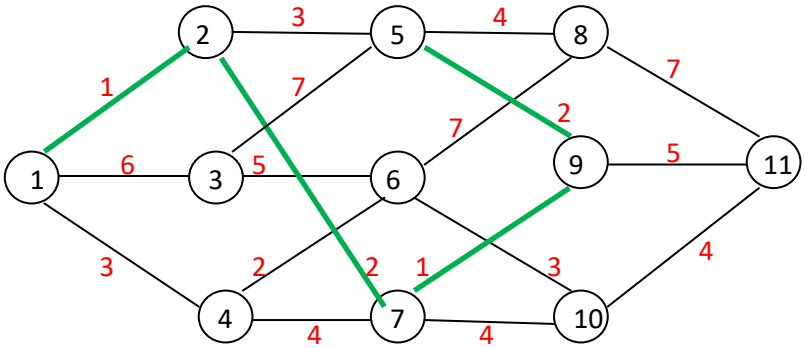
- Мінімальне остове дерево

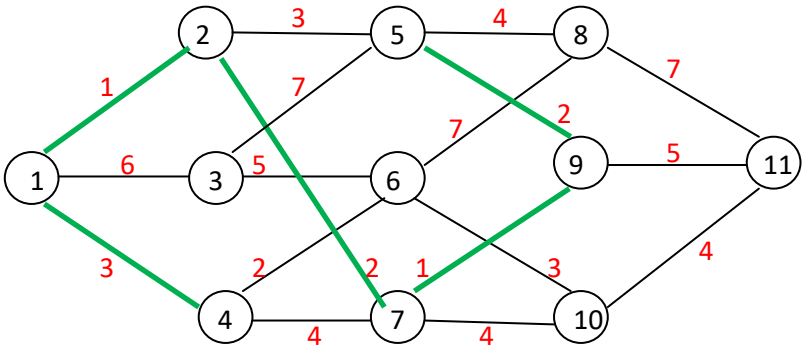
2)Методом Прима:

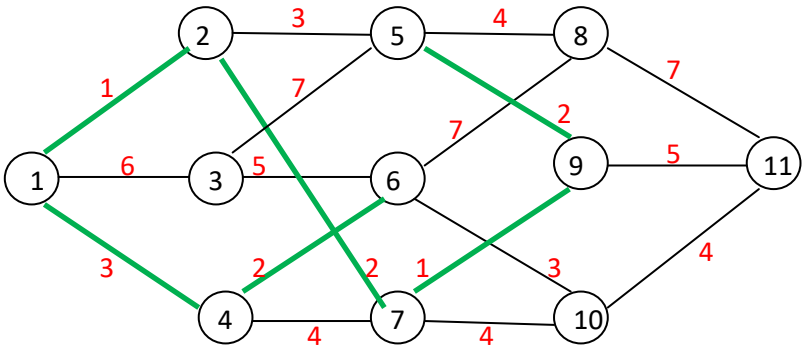


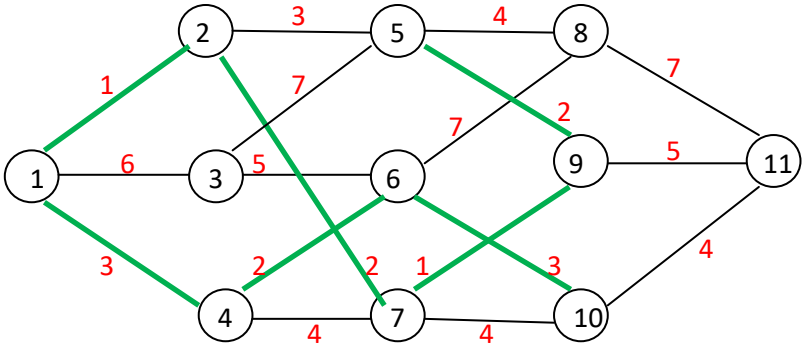

1
2
3
4
5
6
7
8
9
10
11

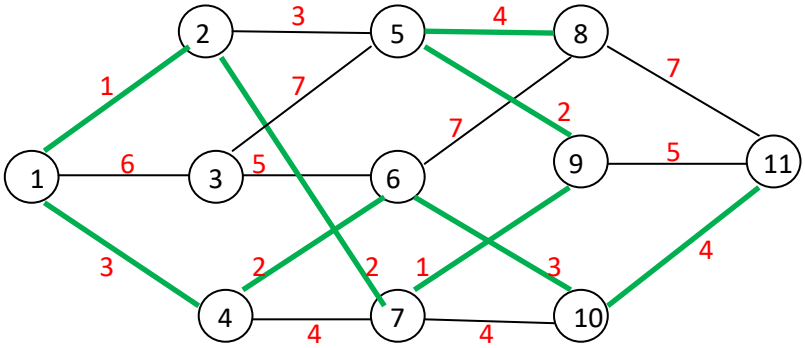

1
2
3
4
5
6
7
8
9
10
11

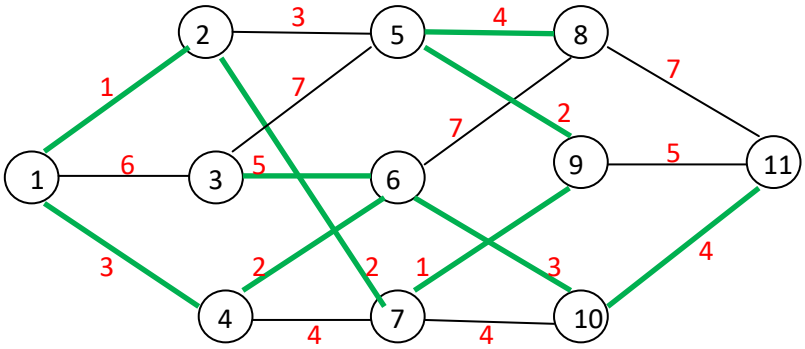

1
2
3
4
5
6
7
8
9
10
11

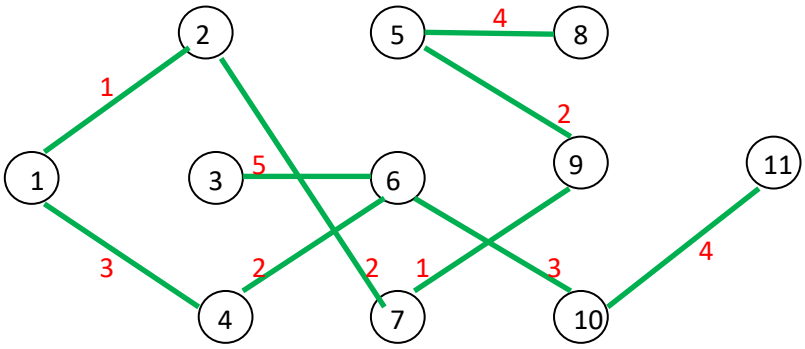

1
2
3
4
5
6
7
8
9
10
11


1
2
3
4
5
6
7
8
9
10
11







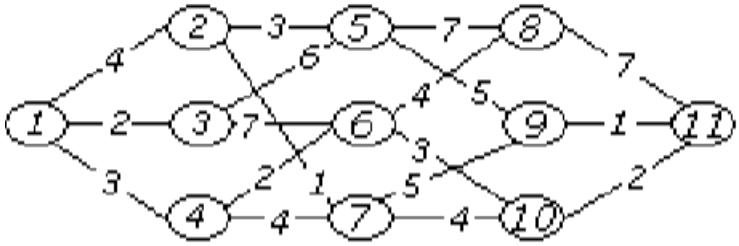


- Мінімальне остове дерево

Завдання №2.

Написати програму, яка реалізує алгоритм знаходження остового дерева мінімальної ваги згідно свого варіанту.

За алгоритмом Краскала знайти мінімальне остове дерево графа. Етапи розв'язання задачі виводити на екран. Протестувати розроблену програму на наступному графі:



Код програми:

```
#include<stdio.h>
#include<stdlib.h>
int i,j,a,b,u,v,n,ne=1;
int min,mincost=0,cost[30][30],parent[30];
int find(int);
int uni(int,int);
void main()
{

    printf("\nEnter the number of vertices:");
    scanf("%d",&n);
    printf("\nEnter the cost adjacency matrix:\n");
    for(i=1;i<=n;i++)
    {
        for(j=1;j<=n;j++)
        {
            scanf("%d",&cost[i][j]);
            if(cost[i][j]==0)
                cost[i][j]=999;
        }
    }
    printf("The edges of Minimum Cost Spanning Tree are\n");
    while(ne < n)
    {
        for(i=1,min=999;i<=n;i++)
        {
            for(j=1;j <= n;j++)
            {
                if(cost[i][j] < min)
                {
                    min=cost[i][j];
                    a=u=i;
                    b=v=j;
                }
            }
        }
        u=find(u);
        v=find(v);
        if(uni(u,v))
        {
            printf("%d edge (%d,%d) =%d\n",ne++,a,b,min);
            mincost +=min;
        }
        cost[a][b]=cost[b][a]=999;
    }
    printf("\n\tMinimum cost = %d\n",mincost);

}
int find(int i)
{
```

```
    while(parent[i])
        i=parent[i];
    return i;
}
int uni(int i,int j)
{
    if(i!=j)
    {
        parent[j]=i;
        return 1;
    }
    return 0;
}
```

Результат виконання програми:

```
Enter the no. of vertices:11
11

Enter the cost adjacency matrix:
0
0
4
4
2
2
3
3
0
0
0
0
0
0
0
0
0
0
0
0
0
0
4
4
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
7
7
1
1
2
2
0
0
The edges of Minimum Cost Spanning Tree are
1 edge (2,7) =1
2 edge (9,11) =1
3 edge (1,3) =2
4 edge (4,6) =2
5 edge (10,11) =2
6 edge (1,4) =3
7 edge (2,5) =3
8 edge (6,10) =3
9 edge (1,2) =4
10 edge (6,8) =4

        Minimum cost = 25
```