

Міністерство освіти і науки, молоді та спорту України  
Національний університет «Львівська політехніка»  
Інститут комп'ютерних наук та інформаційних технологій  
Кафедра систем штучного інтелекту



## **ЗВІТ**

### **Про виконання лабораторної роботи № 4**

*“ Основні операції над графами. Знаходження остова мінімальної ваги за алгоритмом Прима-Краскала ”*

**з дисципліни «Дискретна математика»**

**Виконав:**

студ. групи КН-112

Кадоб'янський І.І.

**Викладач:**

Мельникова

Н.І.

«\_\_» \_\_\_\_\_ 2019 р.

$\Sigma$  = \_\_\_\_

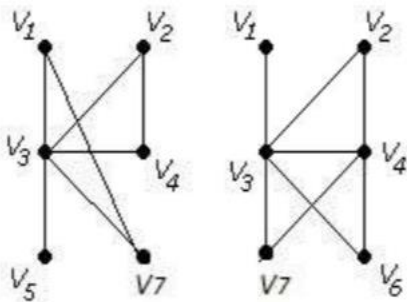
**Мета роботи:** набуття практичних вмінь та навичок з використання алгоритмів Пріма і Краскала.

### Варіант 7

### Індивідуальне завдання

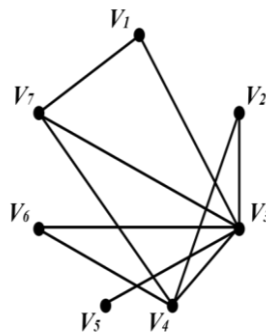
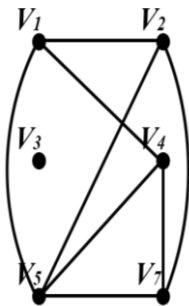
1. Виконати наступні операції над графами:

- 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  $A$ , що складається з 3-х вершин в  $G1$  і знайти стягнення  $A$  в  $G1$  ( $G1 \setminus A$ ),
- 6) добуток графів.



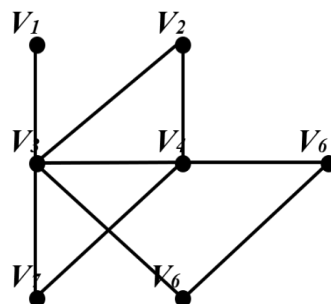
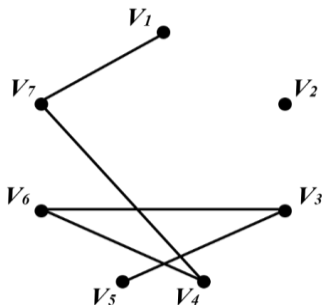
1)

2)

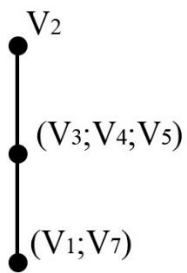


3)

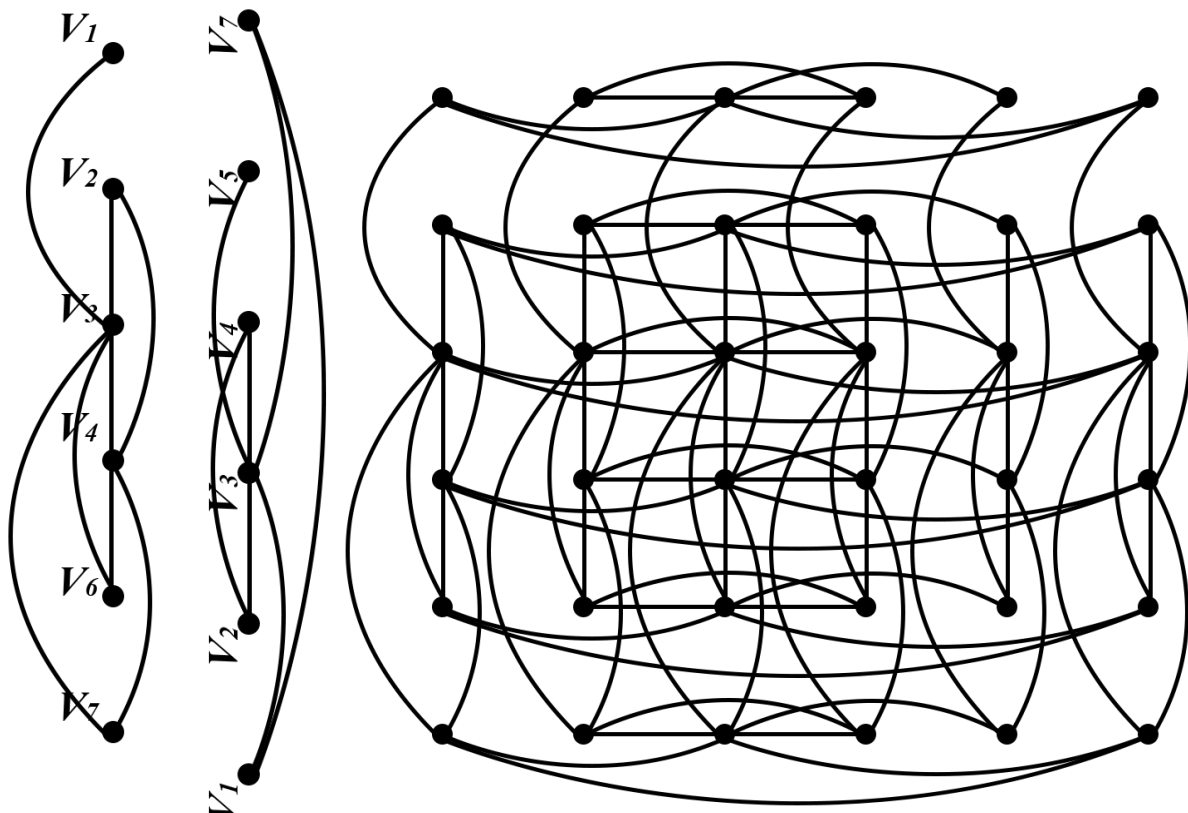
4)



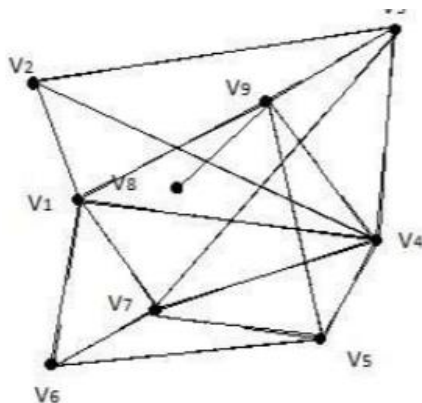
5)



6) Добуток:



2) Знайти таблицю суміжності та діаметр графа.

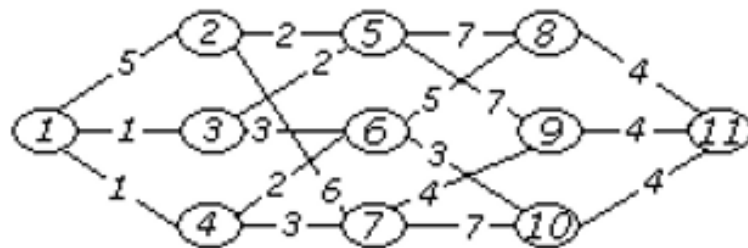


	V1	V2	V3	V4	V5	V6	V7	V8	V9
V1	0	1	0	1	0	1	1	0	1
V2	1	0	1	1	0	0	0	0	0
V3	0	1	0	1	0	0	1	1	1
V4	1	1	1	0	1	0	1	0	1

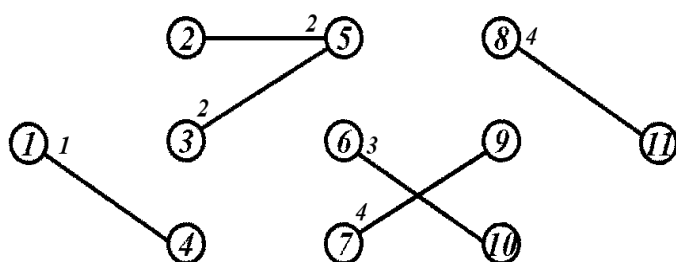
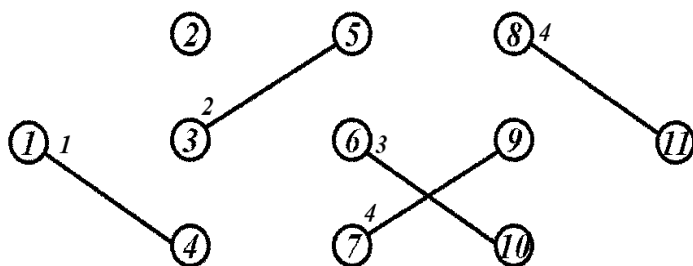
<b>V5</b>	0	0	0	1	0	1	1	0	1
<b>V6</b>	1	0	0	0	1	0	1	0	0
<b>V7</b>	1	0	1	1	1	1	0	0	0
<b>V8</b>	0	0	1	0	0	0	0	0	0
<b>V9</b>	1	0	1	1	1	0	0	0	0

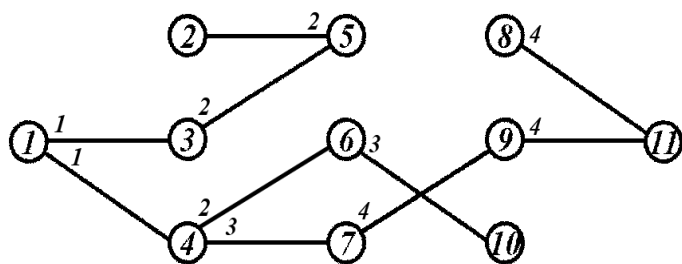
Діаметр графа - 3

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

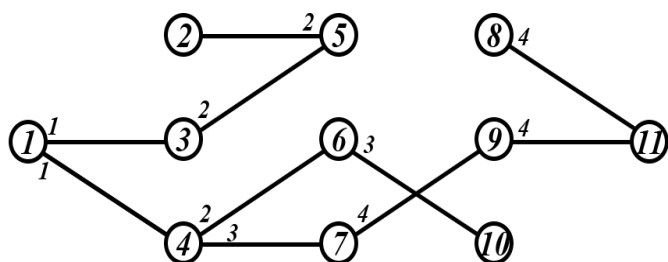
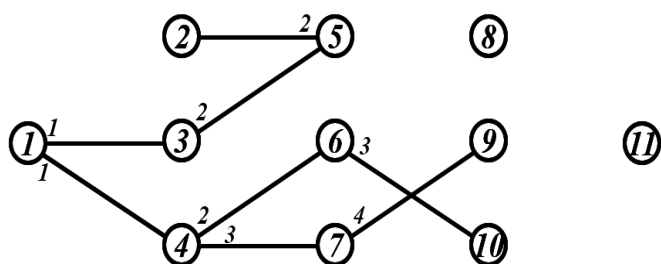
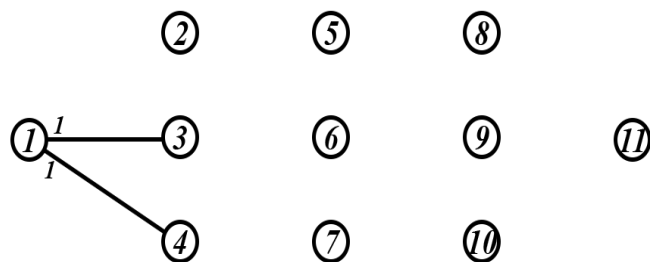


Краскала:



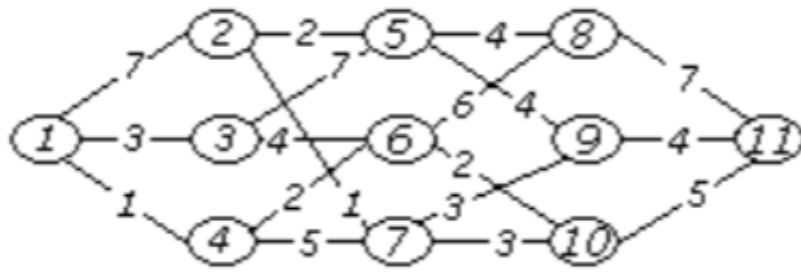


Прима:



## Завдання 2

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



**Код:**

```
#include <iostream>
#include <stdio.h>
#include <stdlib.h>

using namespace std;

int main(void)
{
    int versh, cnt = 0, min_ = 0, n, m;
    bool c = false;

    cout << "The quantity of tops: "; cin >> versh; cout << "\n";

    int **graph = new int*[versh];
    for(int i = 0; i < versh; ++i)
        graph[i] = new int[versh];

    int **rebr = new int*[versh - 1];
    for(int i = 0; i < versh - 1; ++i)
        rebr[i] = new int[2];

    for(int i = 0; i < versh; ++i)
        for (int j = 0; j < versh; ++j)
            cin >> graph[i][j];

    int *tops = new int[versh];
    tops[cnt] = 1;
    ++cnt;

    for(int i = 0; cnt < versh; ++i){
        for(int j = 0; j < cnt; ++j){
            for(int x = 0; x < versh; ++x){
                for(int y = 0; y < cnt; ++y)
                    if(tops[y] == x + 1)
                        c = true;
                if(c == true)
                {
                    c = false;
                    continue;
                }

                if(min_ == 0 && graph[tops[j] - 1][x] > 0)
```

```

        {
            min_ = graph[tops[j] - 1][x];
            n = rebr[cnt - 1][0] = tops[j];
            m = rebr[cnt - 1][1] = x + 1;
            continue;
        }

        if(graph[tops[j] - 1][x] > 0 && graph[tops[j] - 1][x]
< min_)
        {
            min_ = graph[tops[j] - 1][x];
            n = rebr[cnt - 1][0] = tops[j];
            m = rebr[cnt - 1][1] = x + 1;
        }
    }

    graph[n - 1][m - 1] = 0;
    graph[m - 1][n - 1] = 0;
    tops[cnt] = m;
    ++cnt;
    min_ = 0;
}

cout << endl << "Rebra: ";
for (int i = 0; i < versh - 1; ++i)
    cout << "(" << rebr[i][0] << ", " << rebr[i][1] << ") ";}

```

Для вводу:

```

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

```

**Вивід:**

```
"D:\Э\тхЁ\--шёьЁхЄэр ЪрЄхьрЄшър.ý 1\ѡрс 4\ѡрс 4\bin\Debug\ѡрс 4.exe"
The quantity of tops: 11

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

Rebra: (1, 4) (4, 6) (6, 10) (1, 3) (10, 7) (7, 2) (2, 5) (7, 9) (5, 8) (9, 11)
Process returned 0 (0x0)   execution time : 4.116 s
Press any key to continue.
```

**Висновок:**

Я набув практичних вмінь та навичок з використання алгоритмів Прима і Краскала.