

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТУ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

**Кафедра систем штучного інтелекту**

**Лабораторна робота № 4**

з дисципліни

«Дискретна математика»

**Виконала:**

студентка групи КН-112

Сидір Олена Юріївна

**Викладач:**

Мельникова Н.І.

Львів – 2019 р.

**Тема:** 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.

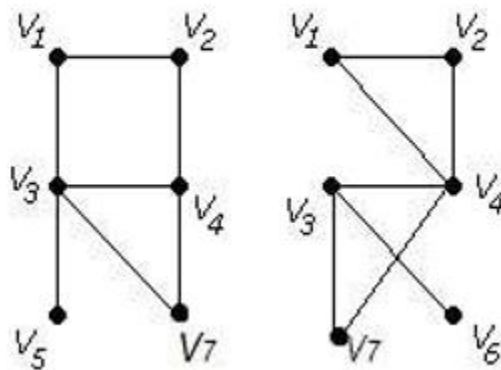
### Варіант №13

#### Завдання № 1.

Розв'язати на графах наступні задачі:

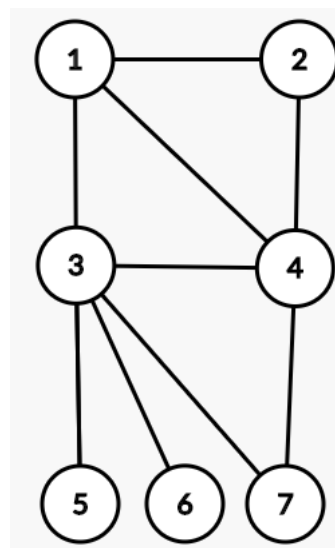
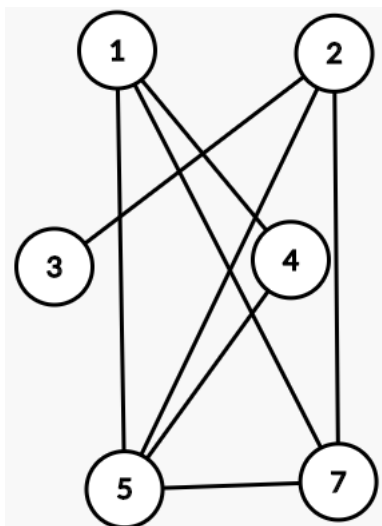
- 1) знайти доповнення до першого графу,
- 2) об'єднання графів,
- 3) кільцеву суму  $G1$  та  $G2$  ( $G1+G2$ ),
- 4) розщепити вершину у другому графі,
- 5) виділити підграф  $A$ , що складається з 3-х вершин в  $G1$  і знайти стягнення  $A$  в  $G1$  ( $G1 \setminus A$ ),
- 6) добуток графів.

13

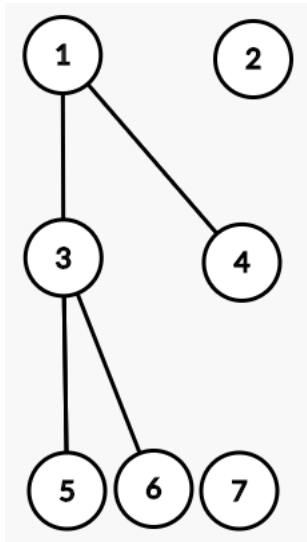


1) Доповнення до 1 графу

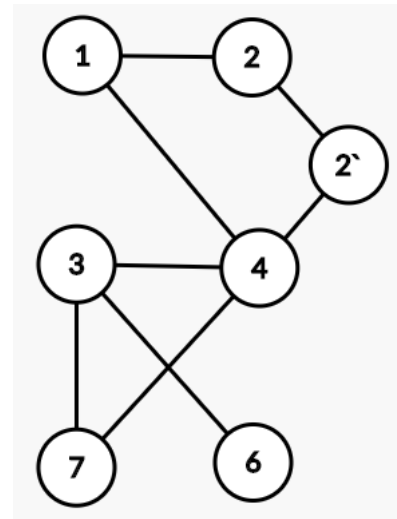
2) Об'єднання графів



3) Кільцева сума  $G1$  і  $G2$ :

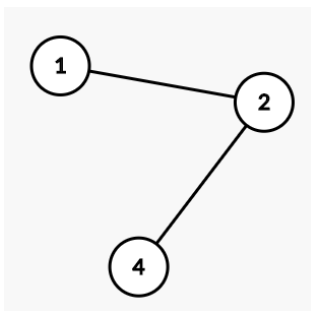


4) Розщепити вершину 2 у  $G2$ :

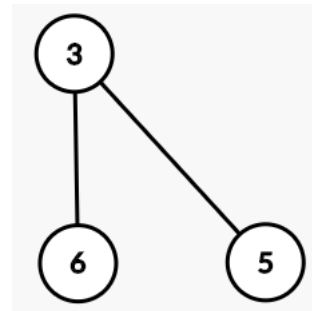


5) Виділити підграф  $A$ , що складається з 3-х вершин в  $G1$  і знайти стягнення  $A$  в  $G1$  ( $G1 \setminus A$ )

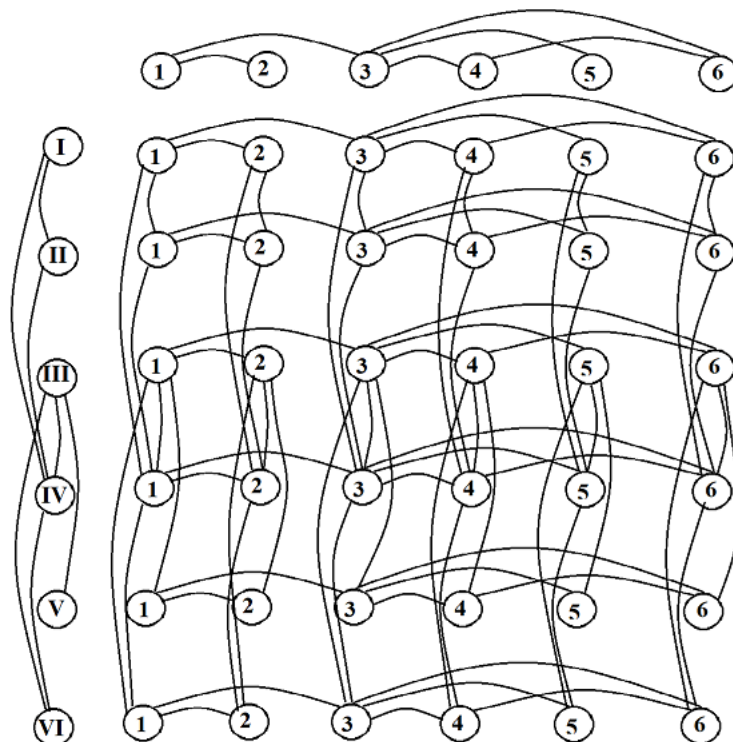
Підграф  $A$ :



$G1$  після стягнення:

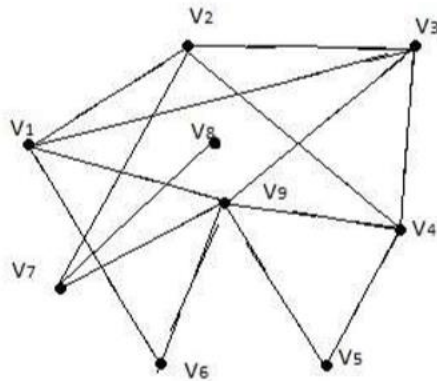


6) Множення графів



2. Знайти таблицю суміжності та діаметр графа.

13



№	1	2	3	4	5	6	7	8	9
1	0	1	1	0	0	1	0	0	1
2	1	0	1	1	0	0	1	0	0
3	1	1	0	1	0	0	0	0	1
4	0	1	1	0	1	0	0	0	1
5	0	0	0	1	0	0	0	0	1
6	1	0	0	0	0	0	0	0	1
7	0	1	0	0	0	0	0	1	1
8	0	0	0	0	0	0	1	0	0
9	1	0	1	1	1	1	1	0	0

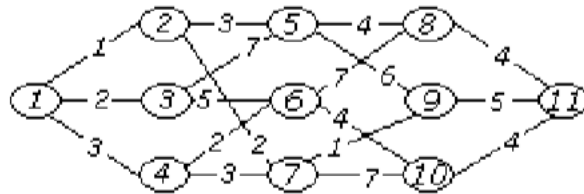
Таблиця суміжності

№	1	2	3	4	5	6	7	8	9
1	0	1	1	2	2	1	2	3	1
2	1	0	1	1	2	2	1	2	2
3	1	1	0	1	2	2	2	3	1
4	2	1	1	0	1	2	2	3	1
5	2	2	2	1	0	2	2	3	1
6	1	2	2	2	2	0	2	3	1
7	2	1	2	2	2	2	0	1	1
8	3	2	3	3	3	3	1	0	2
9	1	2	1	1	1	1	1	2	0

Таблиця інцидентності

Діаметр графа: 3.

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

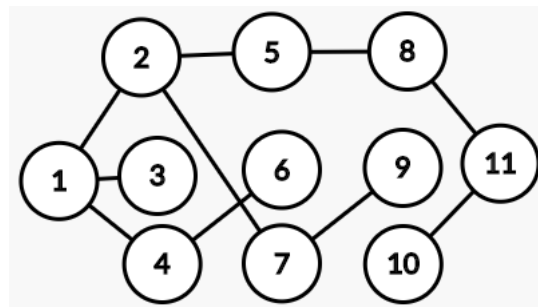
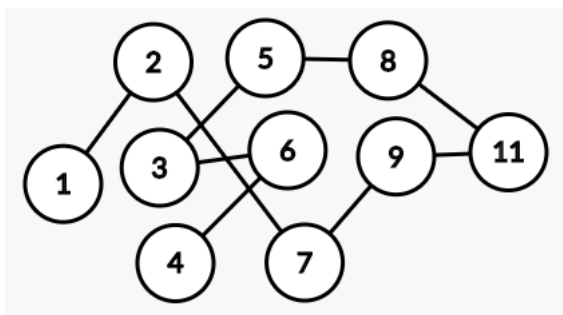


**Прима:**

V	E
1	1,2
3	1,3
2	2,7
7	7,9
9	9,11
11	11,8
8	8,5
5	5,3
3	3,6
6	6,4

**Краскала:**

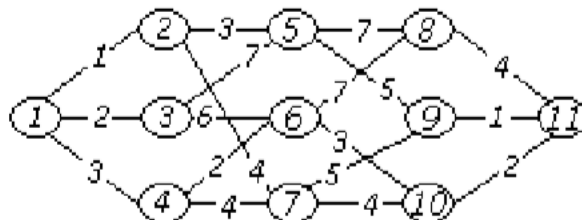
E	Вага
1,2	1
7,9	1
1,3	2
4,6	2
2,7	2
2,5	3
1,4	3
5,8	4
8,11	4
11,10	4



**Завдання 2**

### Варіант № 13

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



№	1	2	3	4	5	6	7	8	9	10	11
1	0	1	2	3	0	0	0	0	0	0	0
2	1	0	0	0	0	0	4	0	0	0	0
3	2	0	0	0	6	6	0	0	0	9	0
4	3	0	0	0	0	2	4	0	0	0	0
5	0	3	6	0	0	0	0	7	5	0	0
6	0	0	6	2	0	0	0	7	6	0	0
7	0	4	0	4	0	0	0	0	5	4	0
8	0	0	0	0	7	7	0	0	0	0	4
9	0	0	0	0	5	0	5	0	0	0	1
10	0	0	0	0	0	3	4	0	0	0	2
11	0	0	0	0	0	0	0	4	1	2	0

**Таблица ваг**

## Програма:

```
#include <bits/stdc++.h>
using namespace std;

#define V 11

int minimalKey(int key[], bool used[])
{
    int min = INT_MAX, min_index;

    for (int v = 0; v < V; v++)
        if (used[v] == false && key[v] < min)
            min = key[v], min_index = v;

    return min_index;
}

void print(int parent[], int graph[V][V])
{
    cout<<"Резултат \tBara\n";
    for (int i = 1; i < V; i++)
        cout<<parent[i]<<" - " <<i<<" \t" <<graph[i][parent[i]]<<" \n";
}

void prima(int graph[V][V])
{
    int parent[V];
    int key[V];
    bool used[V];

    for (int i = 0; i < V; i++) {
        key[i] = INT_MAX;
        used[i] = false;
    }

    key[0] = 0;
    parent[0] = -1;

    for (int count = 0; count < V - 1; count++)
    {
        int u = minimalKey(key, used);
        used[u] = true;
        for (int v = 0; v < V; v++)
            if (graph[u][v] && used[v] == false && graph[u][v] < key[v]) {
```

```

        parent[v] = u;
        key[v] = graph[u][v];
    }

    print(parent, graph);
}

int main()
{
    int graph[V][V] = { { 0, 1, 2, 3, 0, 0, 0, 0, 0, 0, 0 },
                        { 1, 0, 0, 0, 0, 0, 4, 0, 0, 0, 0 },
                        { 2, 0, 0, 0, 6, 6, 0, 0, 0, 9, 0 },
                        { 3, 0, 0, 0, 0, 2, 4, 0, 0, 0, 0 },
                        { 0, 3, 6, 0, 0, 0, 0, 7, 5, 0, 0 },
                        { 0, 0, 6, 2, 0, 0, 0, 7, 0, 0, 0 },
                        { 0, 4, 0, 4, 0, 0, 0, 0, 5, 4, 0 },
                        { 0, 0, 0, 0, 7, 7, 0, 0, 0, 0, 4 },
                        { 0, 0, 0, 0, 5, 0, 5, 0, 0, 0, 1 },
                        { 0, 0, 0, 0, 0, 3, 4, 0, 0, 0, 2 },
                        { 0, 0, 0, 0, 0, 0, 0, 4, 1, 2, 0 }
    };
    prima(graph);
}

```