

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ В  
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА  
ПОЛІТЕХНІКА”**

**Кафедра систем штучного інтелекту**

**Лабораторна робота №4**

з дисципліни  
«Дискретна математика»

**Виконала:**  
студентка групи КН-115  
Дзямба Аліна  
**Викладач:**  
Мельникова Н. І.

Львів – 2019 р.

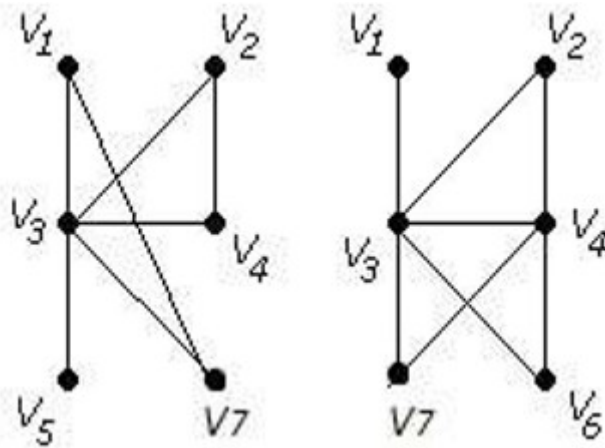
## Основні операції над графами. Знаходження остова мінімальної ваги за алгоритмом Пріма-Краскала

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

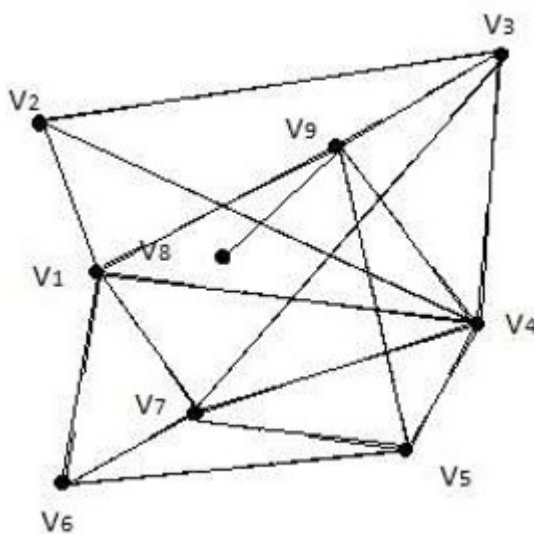
### Варіант №7

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

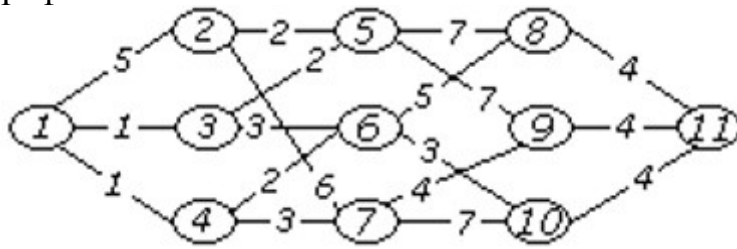
1. Виконати наступні операції над графами:
  - 1) знайти доповнення до першого графу,
  - 2) об'єднання графів,
  - 3) кільцеву суму  $G1$  та  $G2$  ( $G1+G2$ ),
  - 4) розщепити вершину у другому графі,
  - 5) виділити підграф  $A$ , що складається з 3-х вершин в  $G1$  і знайти стягнення  $A$  в 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



2. Знайти таблицю суміжності та діаметр графа:

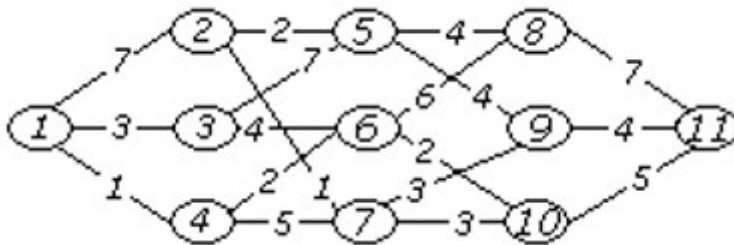


3. Знайти двома методами (Краскала і Прима) мінімальне остове дерево графа:



**Завдання №2.** Написати програму, яка реалізує алгоритм знаходження остового дерева мінімальної ваги згідно свого варіанту.

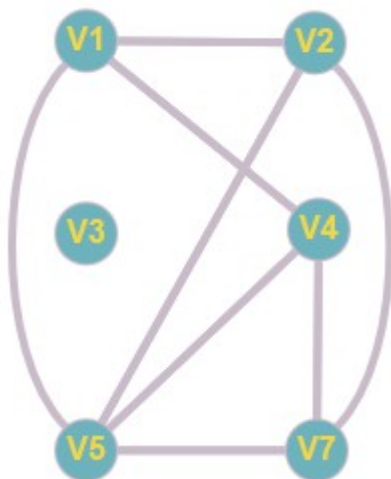
За алгоритмом Прима знайти мінімальне остове дерево графа. Етапи розв'язання задачі виводити на екран. Протестувати розроблену програму на наступному графі:



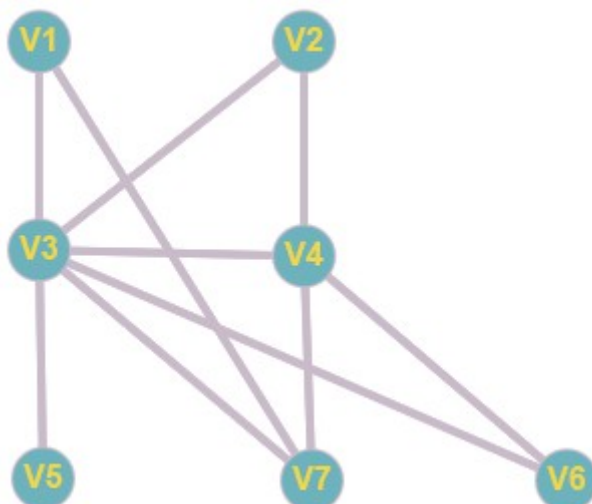
**Розв'язання:**

**Завдання № 1.**

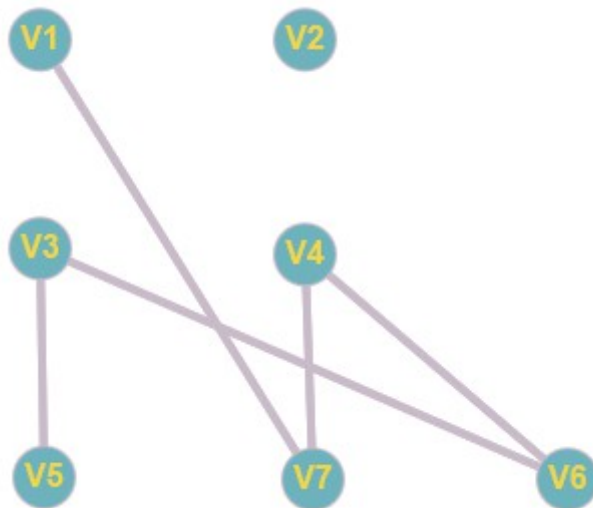
1. 1) знайти доповнення до першого графу



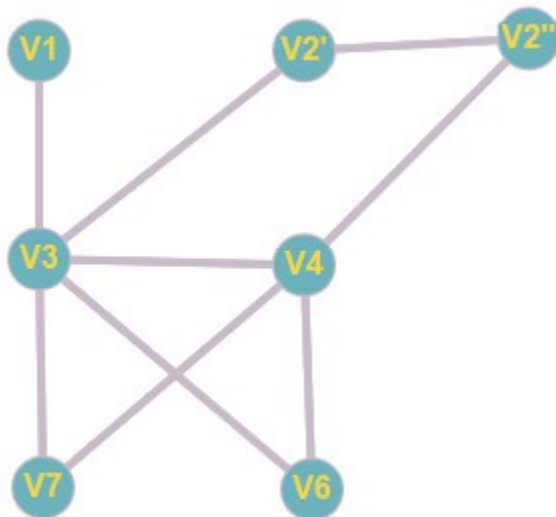
- 2) об'єднання графів



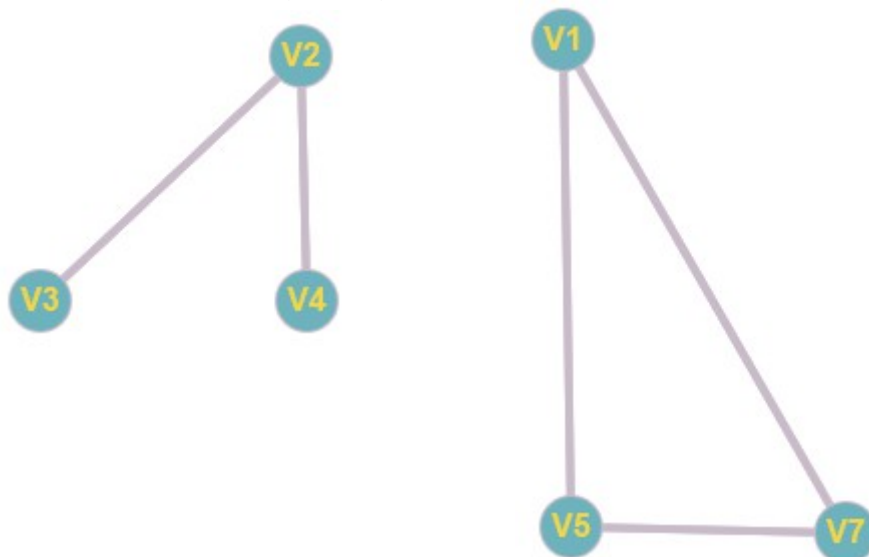
3) кільцеву суму  $G1$  та  $G2$  ( $G1+G2$ )



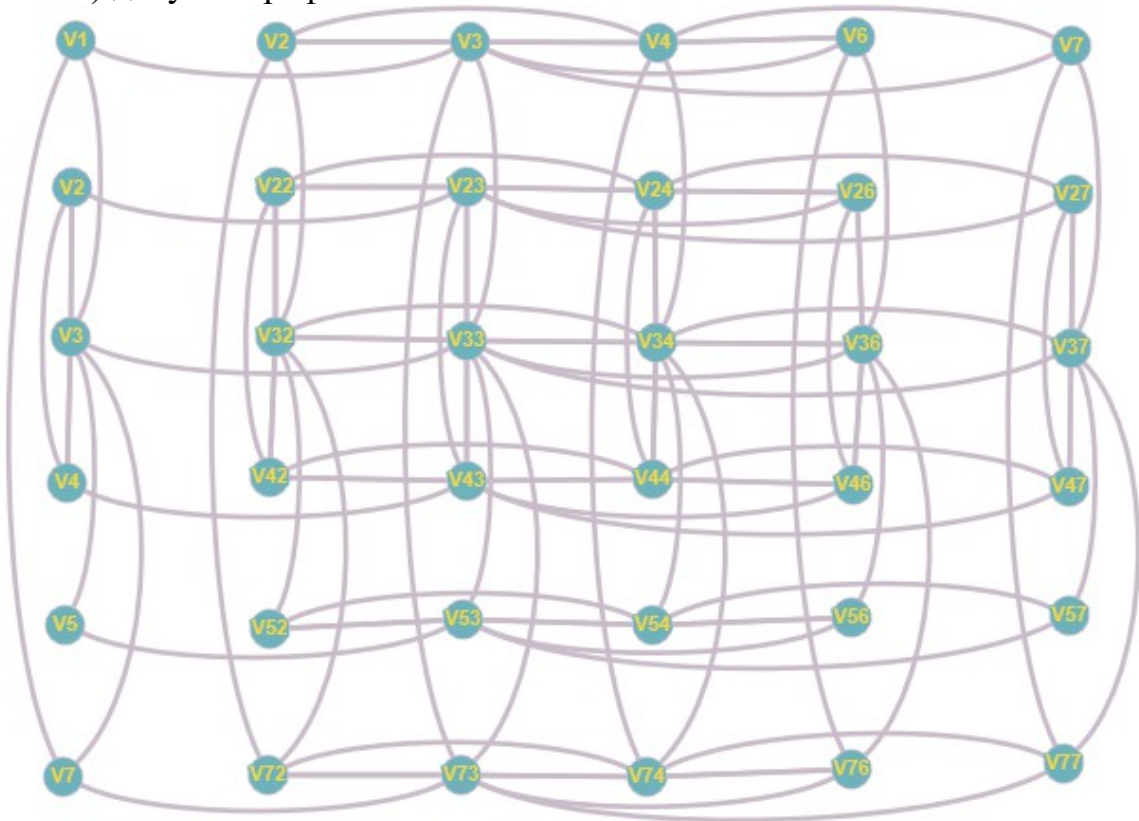
4) розщепити вершину у другому графі  
Розщеплення вершини  $V2$ :



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



б) добуток графів



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  | 0  | 0  | 0  | 0  | 0  | 0  |
| V3 | 0  | 1  | 0  | 1  | 0  | 0  | 1  | 0  | 1  |
| V4 | 1  | 0  | 1  | 0  | 1  | 0  | 1  | 0  | 1  |
| V5 | 0  | 0  | 0  | 1  | 0  | 1  | 1  | 0  | 1  |
| V6 | 1  | 0  | 0  | 0  | 1  | 0  | 1  | 0  | 0  |
| V7 | 1  | 0  | 1  | 1  | 1  | 1  | 0  | 0  | 0  |
| V8 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1  |
| V9 | 1  | 0  | 1  | 1  | 1  | 0  | 0  | 1  | 0  |

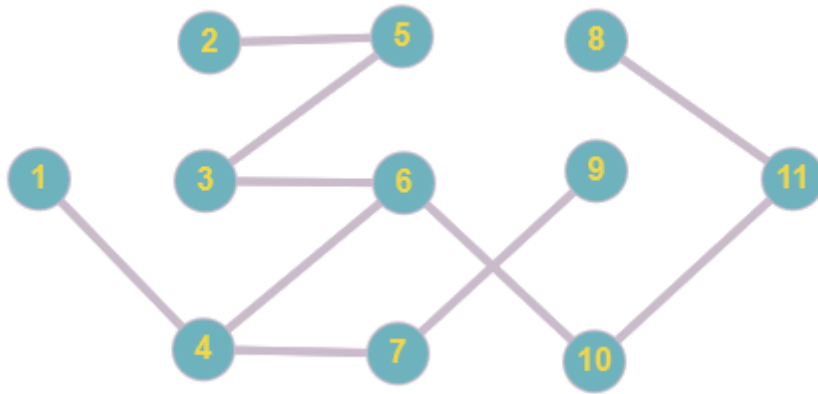
$d(V8; V6) = 3$  – діаметр графа

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

Метод Краскала:

$V = \{3, 5, 6, 2, 1, 4, 7, 10, 9, 11, 8\}$

$E = \{(3; 5), (3; 6), (2; 5), (1; 4), (4; 6), (4; 7), (6; 10), (7; 9), (10; 11), (11; 8)\}$

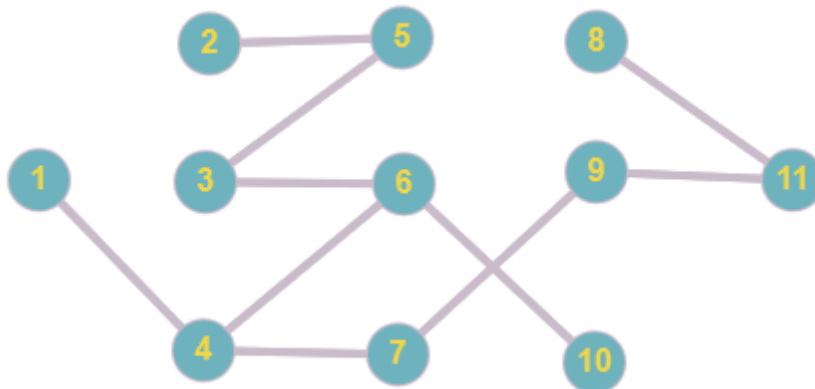


$(1; 3): (1; 4) \rightarrow (4; 6) \rightarrow (6; 3)$   
 $(9; 11): (9; 7) \rightarrow (7; 4) \rightarrow (4; 6) \rightarrow (6; 10) \rightarrow (10; 11)$   
 $(1; 2): (1; 4) \rightarrow (4; 6) \rightarrow (6; 3) \rightarrow (3; 5) \rightarrow (5; 2)$   
 $(5; 8): (5; 3) \rightarrow (3; 6) \rightarrow (6; 10) \rightarrow (10; 11) \rightarrow (11; 8)$   
 $(2; 7): (2; 5) \rightarrow (5; 3) \rightarrow (3; 6) \rightarrow (6; 4) \rightarrow (4; 7)$   
 $(7; 10): (7; 4) \rightarrow (4; 6) \rightarrow (6; 10)$   
 $(6; 8): (6; 10) \rightarrow (10; 11) \rightarrow (11; 8)$   
 $(5; 9): (5; 3) \rightarrow (3; 6) \rightarrow (6; 4) \rightarrow (4; 7) \rightarrow (7; 9)$

Метод Прима:

$V = \{1, 4, 6, 3, 5, 2, 7, 10, 9, 11, 8\}$

$E = \{(1; 4), (4; 6), (6; 3), (3; 5), (5; 2), (4; 7), (6; 10), (7; 9), (9; 11), (11; 8)\}$



**Завдання №2**

```

1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      int n, i, j, k;
6      cout << "Enter the size of the matrix: ";
7      cin >> n;
8      int a[100][100];
9      cout << "Enter the elements of the matrix: \n";
10
11     for (i = 0; i < n; i++)
12     {
13         for (j = 0; j < n; j++)
14         {
15             cin >> a[i][j];
16         }
17     }
18     cout << endl;
19
20     int numb[100]{ 0 };
21     int size = 1;
22     int min_n, min_i, min_j;
23
24     while (size < n)
25     {
26         min_n = 1000;
27
28         for (k = 0; k < size; k++)
29         {
30             for (i = 0; i < n; i++)
31             {
32                 if (a[numb[k]][i] < min_n && a[numb[k]][i] != 0)
33

```

```

34                     min_n = a[numb[k]][i];
35                     min_i = i;
36                     min_j = numb[k];
37                 }
38             }
39         }
40         a[min_j][min_i] = 0;
41         a[min_i][min_j] = 0;
42         bool true_i = 0;
43
44         for (i = 0; i < size; i++)
45             if (min_i == numb[i])
46                 true_i = 1;
47
48         if (true_i == 0)
49         {
50             size++;
51             numb[size - 1] = min_i;
52             cout << "( " << min_j + 1 << ", " << min_i + 1 << " )" << ' ';
53         }
54     }
55 }

```

Результат роботи програми:

```
Enter the size of the matrix: 11
Enter the elements of the matrix:
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 0 4 6 0 0 0 0 7
0 0 0 0 4 0 3 0 0 0 4
0 0 0 0 0 2 3 0 0 0 5
0 0 0 0 0 0 0 7 4 5 0

( 1, 4 );( 4, 6 );( 6, 10 );( 1, 3 );( 10, 7 );( 7, 2 );( 2, 5 );( 7, 9 );( 5, 8 );( 9, 11 );
```

Висновок: Під час виконання лабораторної роботи я отримала практичні вміння та навички з використання алгоритмів Пріма і Краскала.