

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНОМУ  
УНІВЕРСИТЕТУ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**

**Кафедра систем штучного інтелекту**

**Лабораторна робота №4**

з дисципліни  
«Дискретна математика»

**Виконала:**

Студентка групи КН-114

Огорілко Вікторія

**Викладач:**

Мельникова Н.І.

Львів – 2019р.

## Лабораторна робота №4

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

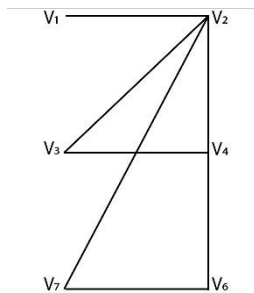
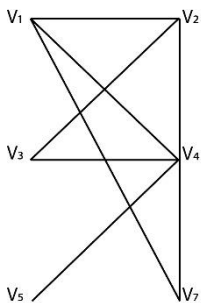
**Мета:** набуття практичних вмінь та навичок з використання алгоритмів Пріма і Краскала.

### Завдання

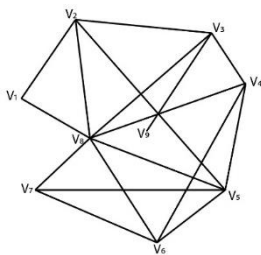
#### Варіант №1

1. Виконати наступні операції над графами:

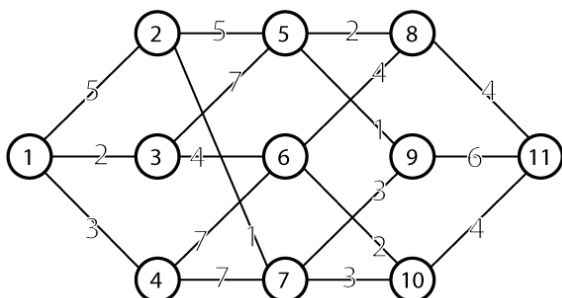
- 1) знайти доповнення до першого графу,
- 2) об'єднання графів,
- 3) кільцеву суму  $G_1$  та  $G_2$  ( $G_1+G_2$ ),
- 4) розщепити вершину у другому графі,
- 5) виділити підграф  $A$ , що складається з 3-х вершин в  $G_1$  і знайти стягнення  $A$  в  $G_1$  ( $G_1 \setminus A$ ),
- 6) добуток графів.



2. Знайти таблицю суміжності та діаметр графа.

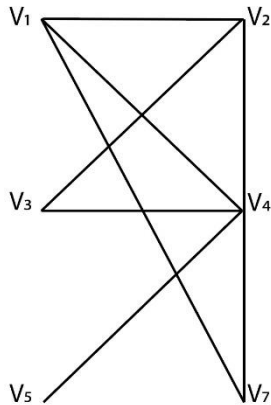


3. Знайти двома методами (Краскала і Пріма) мінімальне остове дерево графа.



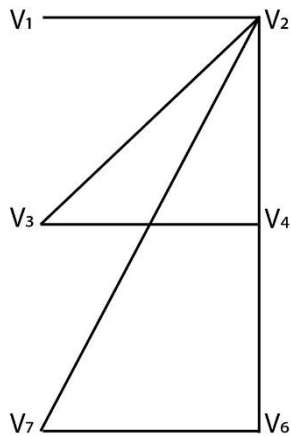
## Розв'язування:

1.



$$X_1 = \{V_1, V_2, V_3, V_4, V_5, V_7\}$$

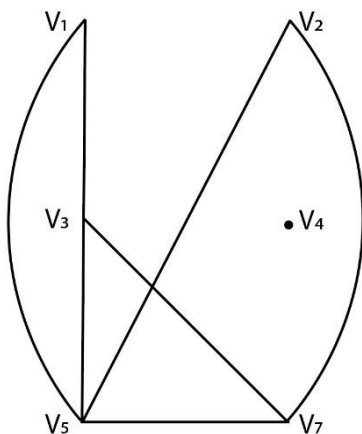
$$W_1 = \{(V_1, V_2), (V_1, V_4), (V_1, V_7), (V_2, V_3), (V_2, V_4), (V_3, V_4), (V_4, V_5), (V_4, V_7)\}$$



$$X_2 = \{V_1, V_2, V_3, V_4, V_6, V_7\}$$

$$W_2 = \{(V_1, V_2), (V_2, V_3), (V_2, V_4), (V_2, V_7), (V_3, V_4), (V_4, V_6), (V_6, V_7)\}$$

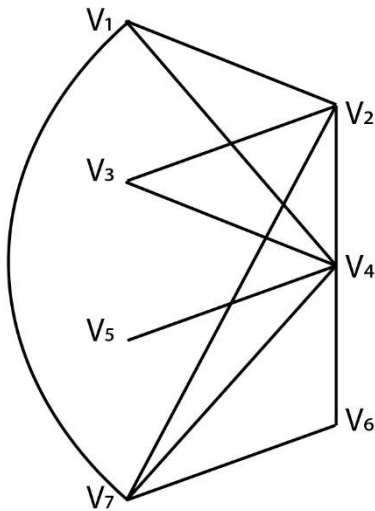
1)Доповнення до першого графу ( $G_3$ ):



$$X_3 = \{V_1, V_2, V_3, V_4, V_5, V_7\}$$

$$W_3 = \{(V_1, V_3), (V_1, V_5), (V_2, V_5), (V_2, V_7), (V_3, V_5), (V_3, V_7), (V_5, V_7)\}$$

2)Об'єднання графів  $G_1$  і  $G_2$  ( $G_1 \cup G_2$ ):



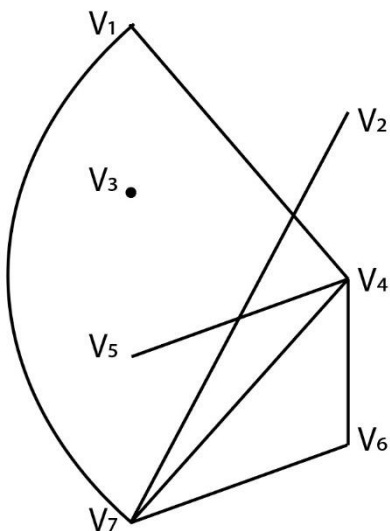
$$X_4 = \{V_1, V_2, V_3, V_4, V_5, V_6, V_7\}$$

$$W_4 = \{(V_1, V_2), (V_1, V_4), (V_1, V_7), (V_2, V_3),$$

$$(V_2, V_4), (V_2, V_7), (V_3, V_4), (V_4, V_5), (V_4, V_6),$$

$$(V_4, V_7), (V_6, V_7)\}$$

3)Кільцева сума  $G_1$  і  $G_2$  ( $G_1 \oplus G_2$ ):



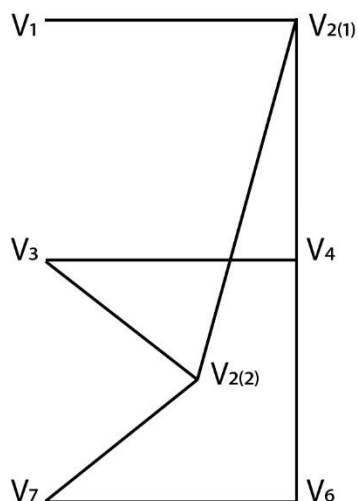
$$X_5 = \{V_1, V_2, V_3, V_4, V_5, V_6, V_7\}$$

$$W_5 = \{(V_1, V_4), (V_1, V_7),$$

$$(V_2, V_7), (V_4, V_5), (V_4, V_6),$$

$$(V_4, V_7), (V_6, V_7)\}$$

4)Розчепимо вершину  $V_2$  на  $V_{2(1)}$  і  $V_{2(2)}$  :

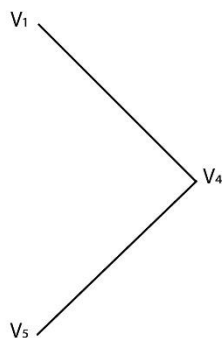


$$X_6 = \{V_1, V_{2(1)}, V_{2(2)}, V_3, V_4, V_6, V_7\}$$

$$W_6 = \{(V_1, V_{2(1)}), (V_{2(1)}, V_{2(2)}), (V_{2(2)}, V_3), (V_{2(1)}, V_4),$$

$$, (V_{2(2)}, V_7), (V_3, V_4), (V_4, V_6), (V_6, V_7)\}$$

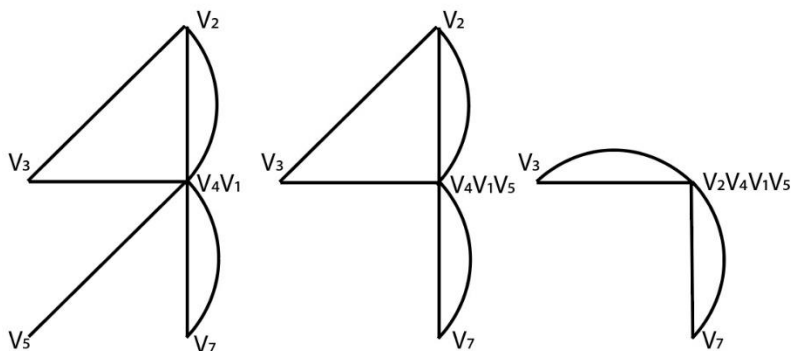
5) Виділимо підграф A :



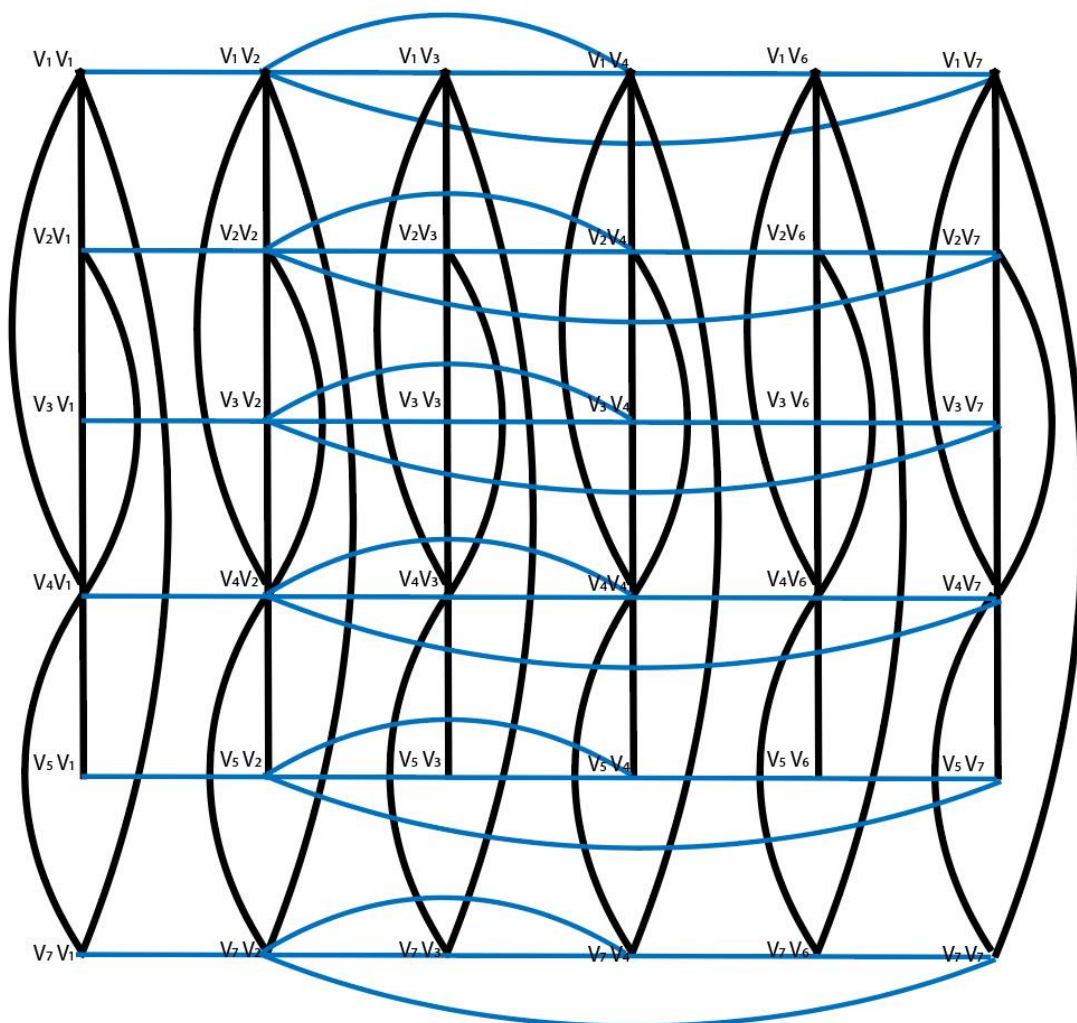
$$X_A = \{V_1, V_4, V_5\}$$

$$W_A = \{(V_1, V_4), (V_4, V_5)\}$$

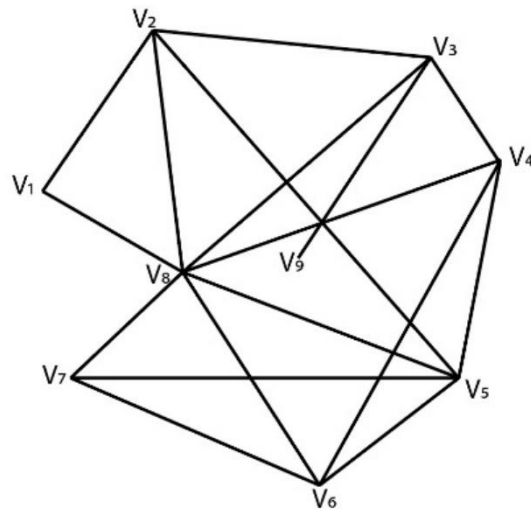
Виконаємо стягнення цього підграфа з графа  $G_1$  ( $G_1 \setminus A$ ):



6) Множення графів  $G_1$  і  $G_2$  ( $G_1 \times G_2$ ):



2.



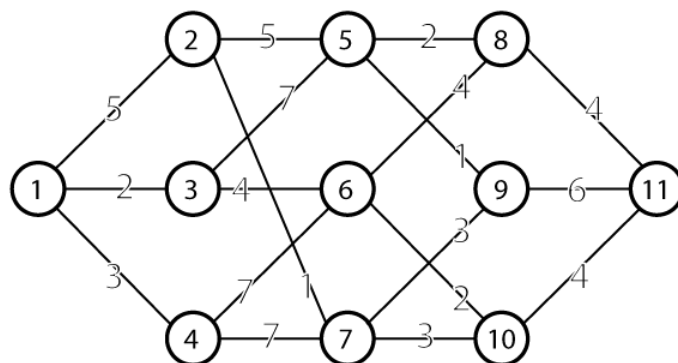
Таблиця інцидентності :

	$V_1$	$V_2$	$V_3$	$V_4$	$V_5$	$V_6$	$V_7$	$V_8$	$V_9$
$V_1$	0	1	0	0	0	0	0	1	0
$V_2$	1	0	1	0	1	0	0	1	0
$V_3$	0	1	0	1	0	0	0	1	1
$V_4$	0	0	1	0	1	1	0	1	0
$V_5$	0	1	0	1	0	1	1	1	0
$V_6$	0	0	0	1	1	0	1	1	0
$V_7$	0	0	0	0	1	1	0	1	0
$V_8$	1	1	1	1	1	1	1	0	0
$V_9$	0	0	1	0	0	0	0	0	0

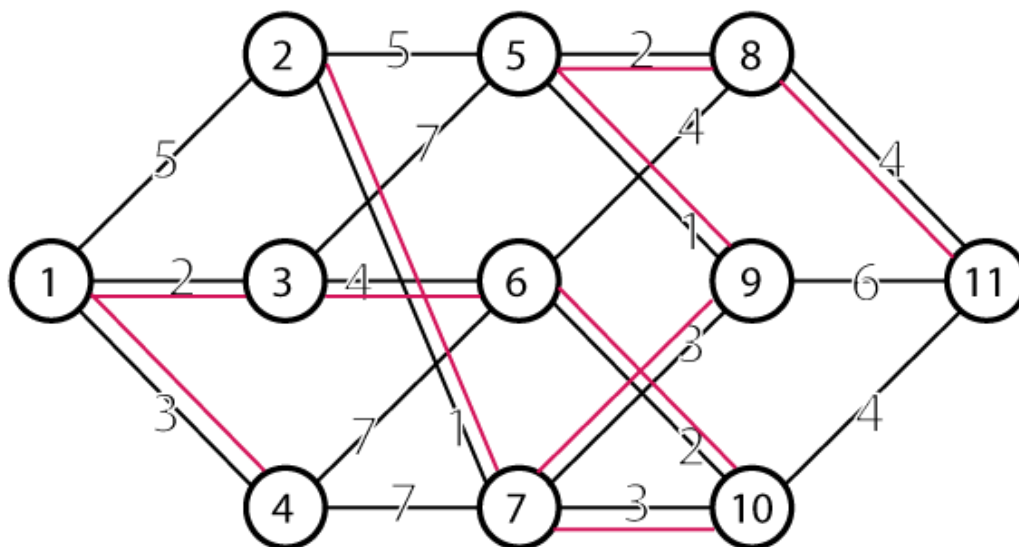
	$V_1$	$V_2$	$V_3$	$V_4$	$V_5$	$V_6$	$V_7$	$V_8$	$V_9$
$V_1$	-	1	2	2	2	2	2	1	3
$V_2$	1	-	1	2	1	2	2	1	2
$V_3$	2	1	-	1	2	2	2	1	1
$V_4$	2	2	1	-	1	1	2	1	2
$V_5$	2	1	2	1	-	1	1	1	3
$V_6$	2	2	2	1	1	-	1	1	3
$V_7$	2	2	2	2	1	1	-	1	3
$V_8$	1	1	1	1	1	1	1	-	2
$V_9$	3	2	1	2	3	3	3	2	-

Діаметр даного графа : 3.

3.

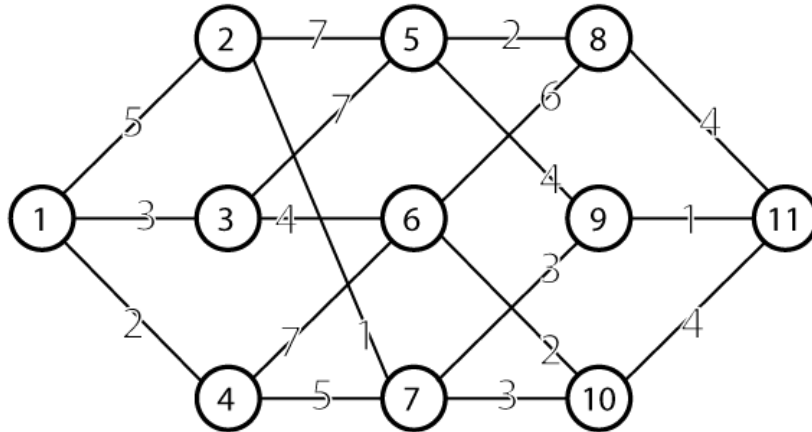


Розв'язок : вага остового дерева =  $2 + 3 + 4 + 2 + 3 + 3 + 1 + 1 + 2 + 4 = 25$



## Додаток 2

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



### Код програми:

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int n;
8      cout << "\nenter number of vertices's : ";
9      cin >> n;
10     int n0;
11     cout << "\nenter number of edges : ";
12     cin >> n0;
13     cout << "\nnow\nenter data in order :\nA(1)  A(2)  A(1)A(2)  (weight of the edge between them)\n";
14     int** arr;
15     arr = new int* [n];
16     for (int i = 0; i < n; i++) {
17         arr[i] = new int[n];
18         for (int j = 0; j < n; j++) {
19             arr[i][j] = 0;
20         }
21     }
22     int a1, a2, w;
23     for (int i = 0; i < n0; i++) {
24         cin >> a1 >> a2 >> w;
25         arr[a1-1][a2-1] = w;
26         arr[a2-1][a1-1] = w;
27     }
28     int* mas;
29     mas = new int[n];
30     for (int i = 0; i < n; i++) {
31         mas[i] = 0;
32     }
33     mas[0] = 1;
34
35     int sum = 0;
36     cout << "\n\n";
37     for (int r=0, t = 0; r != (n - 1);) {
38         t = 0;
39         for (int i = 0; i < n; i++) {
40             if (mas[i] != 0) {
41                 for (int j = 0; j < n; j++) {
42                     if (arr[i][j] != 0) {
43                         if (t == 0) {
44                             a1 = i;
45                             a2 = j;
46                             w = arr[i][j];
47                             t++;
48                         }
49                     }
50                 }
51             }
52         }
53     }
```



```

50         if (arr[i][j] < w) {
51             a1 = i;
52             a2 = j;
53             w = arr[i][j];
54         }
55     }
56 }
57
58 }
59
60 sum += w;
61 cout << "\nA(" << a1 + 1 << " ) -> A(" << a2 + 1 << " ) = " << w;
62 mas[a2] = a2 + 1;
63 arr[a1][a2] = 0;
64 arr[a2][a1] = 0;
65
66 r = 0;
67 for (int q = 0; q < n; q++) {
68     if (mas[q] != 0) { r++; }
69 }
70
71
72 for (int i = 0; i < n; i++) {
73     if (mas[i] == 0) { a2 = i; }
74 }
75 for (int i = 0, t = 0; i < n; i++) {
76     if (arr[i][a2] != 0) {
77         if (t == 0) {
78             a1 = i;
79             w = arr[i][a2];
80             t++;
81         }
82         else {
83             if (arr[i][a2] < w) {
84                 a1 = i;
85                 w = arr[i][a2];
86             }
87         }
88     }
89 }
90 sum += w;
91 cout << "\nA(" << a1 + 1 << " ) -> A(" << a2 + 1 << " ) = " << w;
92 cout << "\n\nminimal weight : " << sum;
93 cout << "\n\n\n";
94 return 0;
95 }

```

### Вивід програми:

```
enter number of vertices's : 11
enter number of edges : 18
now
enter data in order :
A(1)  A(2)  A(1)A(2)  (weight of the edge between them)
1 2 5
1 3 3
1 4 2
2 5 7
2 7 1
3 5 7
3 6 4
4 6 7
4 7 5
5 8 2
5 9 4
8 11 4
6 8 6
6 10 2
7 10 3
7 9 3
9 11 1
10 11 4

A(1) -> A(4) = 2
A(1) -> A(3) = 3
A(3) -> A(6) = 4
A(6) -> A(10) = 2
A(10) -> A(7) = 3
A(7) -> A(2) = 1
A(7) -> A(9) = 3
A(9) -> A(11) = 1
A(9) -> A(5) = 4
A(5) -> A(8) = 2

minimal weight : 25
```

**Висновок:** Я отримала практичні вміння та навички з використання алгоритмів Прима і Краскала.