

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

**Кафедра систем штучного інтелекту**

**Лабораторна робота №4**

з дисципліни  
«Дискретна математика»

**Виконав:**  
студент групи КН-113  
Добосевич Данило

**Викладач:**  
Мельникова Н.І.

Львів – 2019 р.

## Лабораторна робота № 4.

### Варіант №8

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

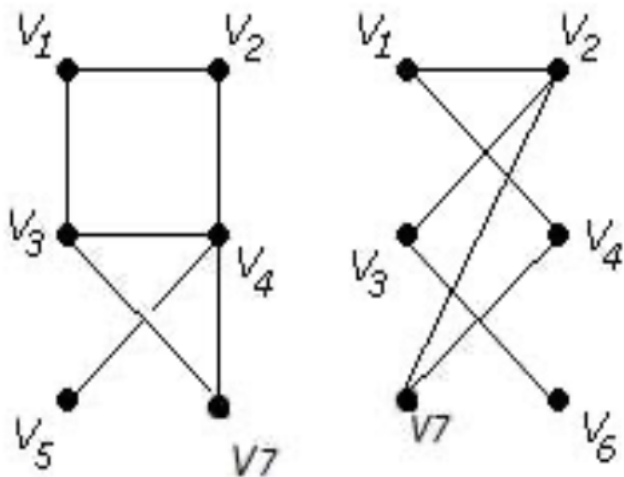
**Мета роботи:** набуття практичних вмінь та навичок з використанням алгоритмів Пріма і Краскала.

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

1. Виконати наступні операції над графами:

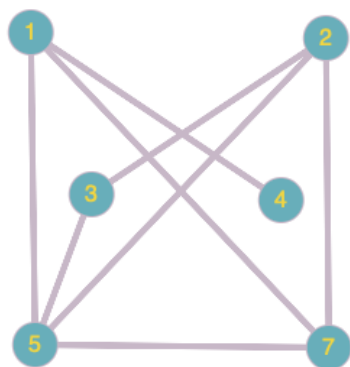
- 1) знайти доповнення до першого графу,
- 2) об'єднання графів,
- 3) кільцеву суму  $G1$  та  $G2$  ( $G1+G2$ ),
- 4) розщепити вершину у другому графі,
- 5) виділити підграф  $A$ , що складається з 3-х вершин в  $G1$  і знайти стягнення  $A$  в  $G1$  ( $G1 \setminus A$ ),
- 6) добуток графів.

8

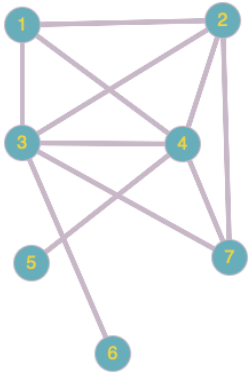


**Розв'язки:**

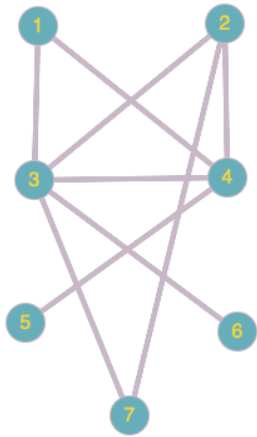
1) знайти доповнення до першого графу



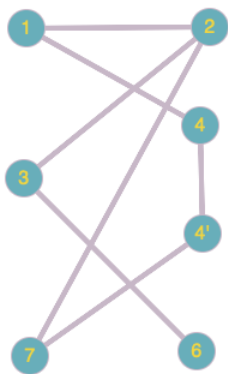
2) об'єднання графів,



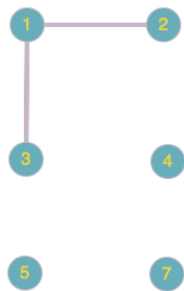
3) кільцеву суму  $G_1$  та  $G_2$  ( $G_1 \oplus G_2$ ),



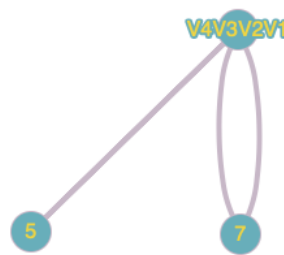
4) розщепити вершину у другому графі,



5) виділити підграф  $A$ , що складається з 3-х вершин в  $G_1$  і знайти стягнення  $A$  в  $G_1$  ( $G_1 \setminus A$ ),

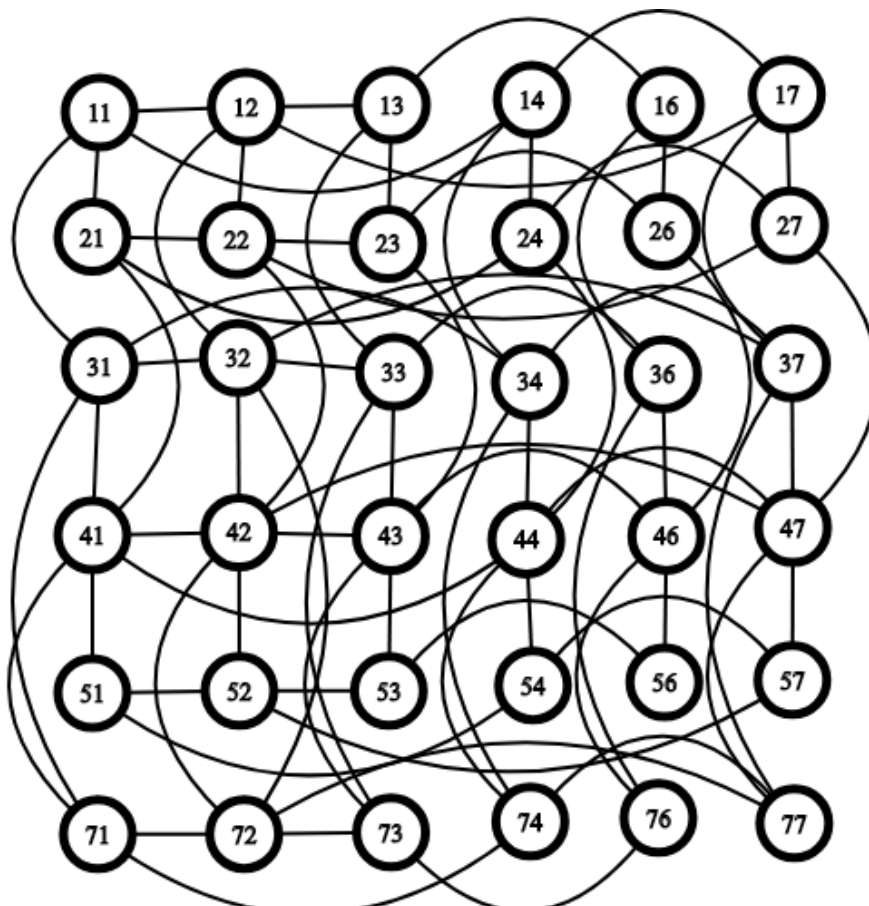


Підграф  $A$



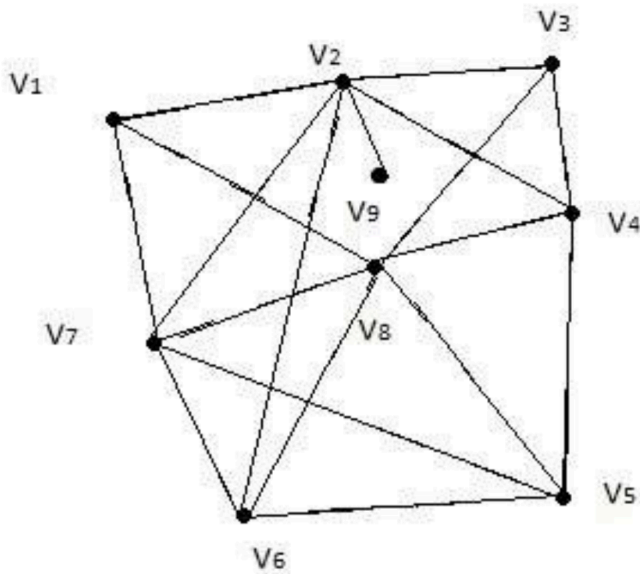
Стягнення  $A$  в  $G_1$

6) добуток графів.



2. Знайти таблицю суміжності та діаметр графа.

8



Діаметр:  $d = 3(V5-V4-V2-V9)$

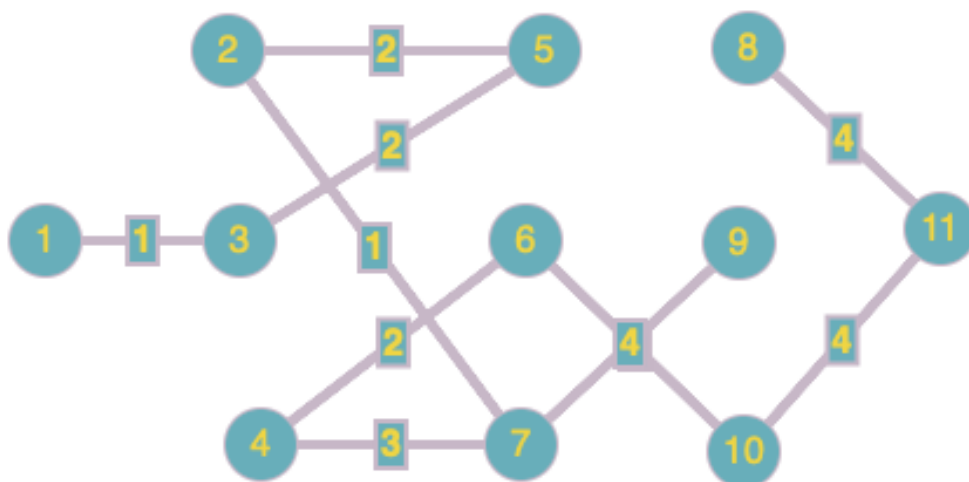
	V1	V2	V3	V4	V5	V6	V7	V8	V9
V1	0	1	0	0	0	0	1	1	0
V2	1	0	1	1	0	1	1	0	1
V3	0	1	0	1	0	0	0	1	0
V4	0	0	1	0	1	0	0	1	0
V5	0	0	0	1	0	1	1	1	1
V6	0	1	0	0	1	0	1	1	0
V7	1	1	0	0	1	1	0	1	0
V8	1	0	1	1	1	1	1	0	0
V9	0	1	0	0	0	0	0	0	0

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

8

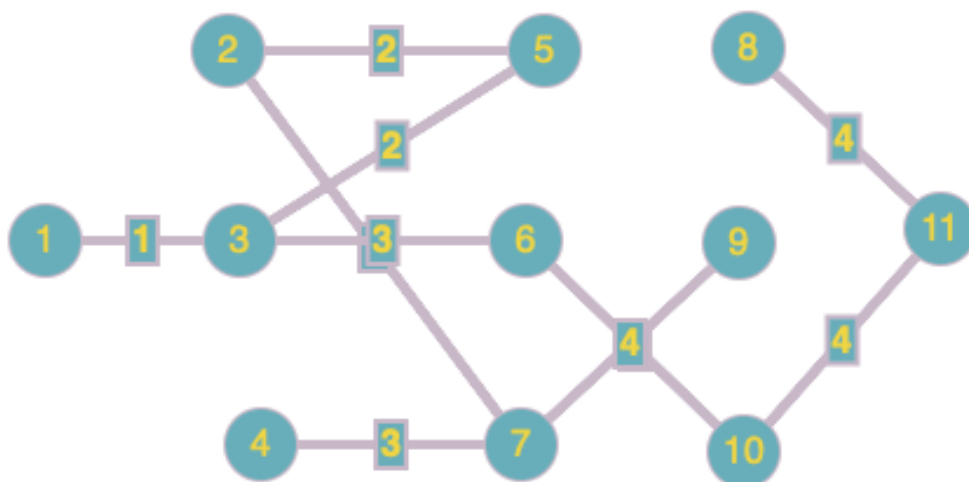


## Алгоритм Краскала



1-3 (1) -> 2-7 (1) -> 2-5 (2) -> 3-5 (2) -> 4-6 (2) -> 4-7 (3) -> 6-10 (3) -> 10-11 (4) -> 8-11 (4) -> 7-9 (4) = 26.

## Алгоритм Прима

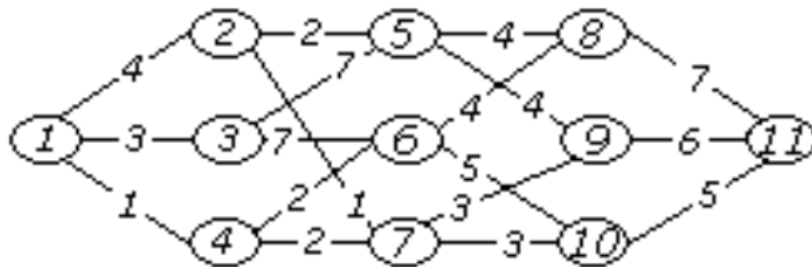


1-3 (1) -> 3-5 (2) -> 5-2 (2) -> 2-7(1)->3-6 (3) -> 7-4 (3) -> 6-10 (3) -> 10-11 (4) -> 11-8 (4) -> 7-9 (4) = 27.

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

### Варіант № 8

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



Програма:

```

1  #include <stdio.h>
2  #define a_rebr 18
3  #define a_top 11
4  #define c_1 4
5  #define c_2 2
6
7  int main (void)
8  {
9      int i,j,k,a;
10     int rebro [a_rebr][c_1] =
11     {
12         {1, 2, 4, 0}, //point from, point to, weight, +/- in tree
13         {1, 3, 3, 0},
14         {1, 4, 1, 0},
15         {2, 5, 2, 0},
16         {2, 7, 1, 0},
17         {3, 5, 7, 0},
18         {3, 6, 7, 0},
19         {4, 6, 2, 0},
20         {4, 7, 2, 0},
21         {5, 8, 4, 0},
22         {5, 9, 4, 0},
23         {6, 8, 4, 0},
24         {6, 10, 5, 0},
25         {7, 9, 3, 0},
26         {7, 10, 3, 0},
27         {8, 11, 7, 0},
28         {9, 11, 6, 0},
29         {10, 11, 5, 0}
30     };
31
32     int top [a_top][c_2] =
33     {
34         {1, 0},
35         {2, 0},
36         {3, 0},
37         {4, 0},
38         {5, 0},
39         {6, 0},
40         {7, 0},
41         {8, 0},

```

```

42         {9, 0},
43         {10, 0},
44         {11, 0}
45     };
46
47     //sort
48     for (i = 0; i < a_rebr; i++)
49     {
50         for (j = 0; j < a_rebr - i - 1; j++)
51         {
52             if (rebro [j][2] > rebro [j + 1][2])
53             {
54                 for (k = 0, a = 0; k < c_1; k++)
55                 {
56                     a = rebro [j + 1][k];
57                     rebro [j + 1][k] = rebro [j][k];
58                     rebro [j][k] = a;
59                 }
60             }
61         }
62     }
63
64     int count = 1;
65     for (i = 0; i < a_rebr; i++) {
66         if (top [rebro[i][0] - 1][1] != top [rebro[i][1] - 1][1]) {
67             rebro [i][3] = 1;
68             if (top [rebro[i][0] - 1][1] == 0 && top [rebro[i][1] - 1][1] != 0)
69                 top [rebro[i][0] - 1][1] = top [rebro[i][1] - 1][1];
70             if (top [rebro[i][0] - 1][1] != 0 && top [rebro[i][1] - 1][1] == 0)
71                 top [rebro[i][1] - 1][1] = top [rebro[i][0] - 1][1];
72             if (top [rebro[i][0] - 1][1] != 0 && top [rebro[i][1] - 1][1] != 0)
73                 for (j = 0; j < a_top; j++)
74                 {
75                     if (top [j][1] == top [rebro[i][1] - 1][1])
76                     {
77                         top [j][1] = top [rebro[i][0] - 1][1];
78                     }
79                 }
80             }
81         }
82     }
83     }
84     if (top [rebro[i][0] - 1][1] == 0 && top [rebro[i][1] - 1][1] == 0)
85     {
86         rebro [i][3] = 1;
87         top [rebro[i][0] - 1][1] = count;
88         top [rebro[i][1] - 1][1] = count;
89         count++;
90     }
91 }
92
93 for (i = 0; i < a_rebr; i++)
94 {
95     if (rebro[i][3] == 1)
96     {
97         printf ("%i - %i (%i)\n", rebro[i][0], rebro[i][1], rebro [i][2]);
98     }
99 }
100
101 return 0;
102 }

```

**Висновок:** на цій лабораторній ми набули практичних вмінь та навичок з використанням алгоритмів Пріма і Краскала.



