

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

Кафедра систем штучного інтелекту

**Лабораторна робота №\_4**

з дисципліни  
«Дискретна математика»

**Виконала:**

Студентка  
групи КН-112

Гудз Юлія

**Викладач:**

Мельникова Н.І.

Львів – 2019 р.

## Лабораторна робота № 4.

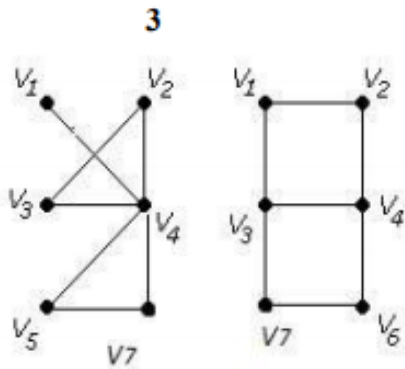
**Тема:** Основні операції над графами. Знаходження остова мінімальної ваги за алгоритмом Пріма-Краскала

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

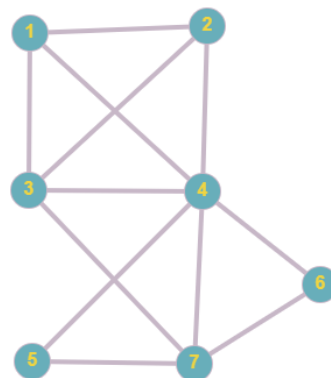
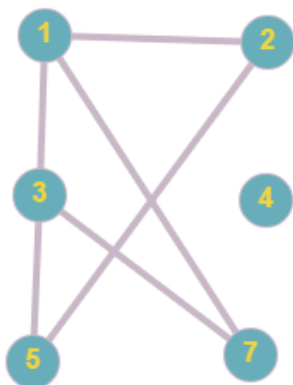
### Варіант № 3

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

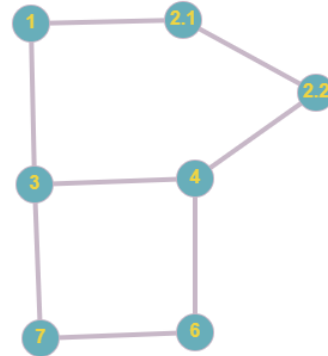
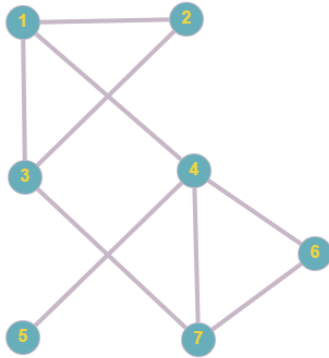
1. Виконати наступні операції над графами:



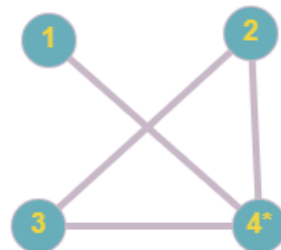
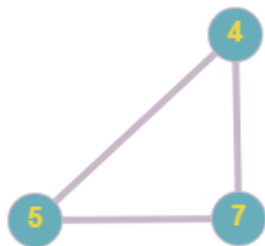
1) знайти доповнення до першого графу. 2) об'єднання графів



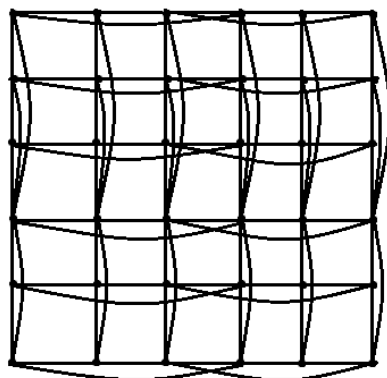
3) кільцеву суму  $G1$  та  $G2$  ( $G1+G2$ ) 4) розщепити вершину у другому графі



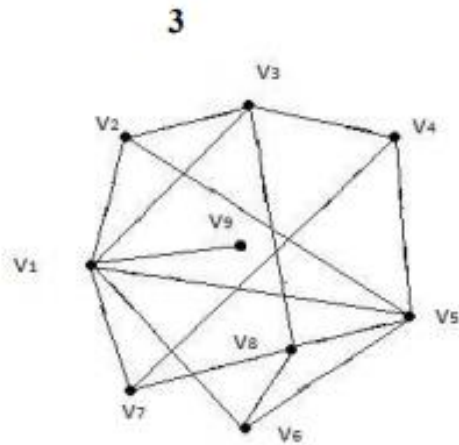
5) виділити підграф  $A$ , що складається з 3-х вершин в  $G1$  і знайти стягнення  $A$  в  $G1$  ( $G1 \setminus A$ )



6) добуток графів.



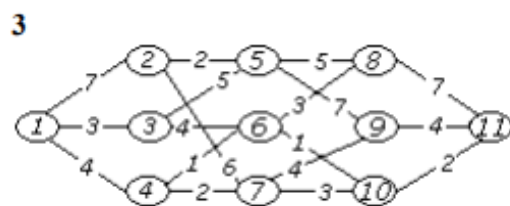
2. Знайти таблицю суміжності та діаметр графа.



	1	2	3	4	5	6	7	8	9
1	0	1	1	0	1	1	1	0	1
2	1	0	1	0	1	0	0	0	0
3	1	1	0	1	0	0	0	1	0
4	0	0	1	0	1	0	1	0	0
5	1	1	0	1	0	1	0	1	0
6	1	0	0	0	1	0	0	1	0
7	1	0	0	1	0	0	0	1	0
8	0	0	1	0	1	1	1	0	0
9	1	0	0	0	0	0	0	0	0

Діаметр=3

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



Метод Прима:

(1,3)=3

(3,6)=4

(6,10)=1

(6,4)=1

$$(4, 7)=2$$

$$(10,11)=2$$

$$(6,8)=3$$

$$(7,9)=4$$

$$(3,5)=5$$

$$(5,2)=2$$

$$3+4+1+1+2+2+3+4+5+2=27$$

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

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

Метода Краскала:

$$(4,6)=1$$

$$(6,10)=1$$

$$(4,7)=2$$

$$(10,11)=2$$

$$(2,5)=2$$

$$(1,3)=3$$

$$(6,8)=3$$

$$(11,9)=4$$

$$(3,6)=4$$

$$(5,8)=5$$

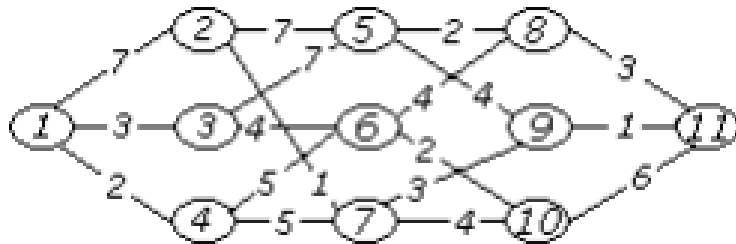
$$1+1+2+2+2+3+3+4+4+5=27$$

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

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

**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:



**Програмна реалізація:**

```

1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      setlocale(LC_ALL, "Ukrainian");
7      int v, N = 0;
8      int MIN = 0, x, l;
9      v=11;
10     int matrix[v][v]
11     {{0,7,5,2,0,0,0,0,0,0,0},
12      {7,0,0,0,7,0,5,0,0,0,0},
13      {5,0,0,0,7,4,0,0,0,0,0},
14      {2,0,0,0,0,3,1,0,0,0,0},
15      {0,7,7,0,0,0,0,4,0,0,0},
16      {0,0,4,3,0,0,0,0,4,0,2},
17      {0,5,0,1,0,0,0,0,3,2,0},
18      {0,0,0,0,4,0,0,0,0,0,3},
19      {0,0,0,0,4,0,3,0,0,0,6},
20      {0,0,0,0,0,2,2,0,0,0,1},
21      {0,0,0,0,0,0,0,3,6,1,0}
22     };
23     int reb[v-1][2];
24     int verish[v];
25
26     int verish[v];
27     bool check = false;
28     cout<<"Матриця:"<<endl;
29     for (int i = 0; i < v; i++) {
30         cout<<endl;
31         for (int j = 0; j < v; j++) {
32             cout<<matrix[i][j]<<" ";
33         }
34         cout<<endl;
35     }
36     verish[N] = 1;
37     N++;
38
39     for (int i = 0; N < v; i++) {
40         for (int j = 0; j < N; j++) {
41             for (int a = 0; a < v; a++) {
42                 for (int m = 0; m < N; m++) {
43                     if (verish[m] == a + 1) {
44                         check = true;
45                     }
46                 }
47             }
48         }
49     }

```

```
main.cpp
48     if (check)
49     { check = false;
50       continue; }
51     if (MIN == 0 && matrix[versh[j] - 1][a] > 0) {
52       MIN = matrix[versh[j] - 1][a];
53       r = reb[N - 1][0] = versh[j];
54       l = reb[N - 1][1] = a + 1;
55       continue;
56     }
57
58     if (matrix[versh[j] - 1][a] > 0 && matrix[versh[j] - 1][a] < MIN) {
59       MIN = matrix[versh[j] - 1][a];
60       r = reb[N - 1][0] = versh[j];
61       l = reb[N - 1][1] = a + 1;
62     }
63   }
64   versh[N] = 1;
65   N++;
66   MIN = 0;
67 }
68 cout<<endl<< "V: { ";
69
70 for (int j = 0; j < v; j++) {
71
72   cout << versh[j]<<" ";
73 }
74
75   cout << " ";
76   cout <<endl<< "E:{ ";
77
78   for (int j = 0; j < v - 1; j++) {
79     cout << "(" << reb[j][0]<<","<< reb[j][1]<<"), ";
80   }
81   cout << "}"<<endl;
82   cout<<endl<<"Ваги ребер:"<<endl;
83   int sum=0;
84   for (int j=0;j<v-1; j++){
85     int k,m;
86     k=reb[j][0];
87     m=reb[j][1];
88     cout<<matrix[k-1][m-1]<<" ";
89     sum+=matrix[k-1][m-1];
90   }
91   cout<<endl<<endl<<"Мінімальне остове дерево: "<<sum<<endl<<endl;
92
93   return 0;
94 }
95
```

## Результат програми:

```
C:\Users\user\Labs\laba.4.dm\bin\Debug\laba.4.exe
Матриця:
0 7 5 2 0 0 0 0 0 0
7 0 0 0 7 0 5 0 0 0
5 0 0 0 7 4 0 0 0 0
2 0 0 0 0 3 1 0 0 0
0 7 7 0 0 0 0 4 0 0
0 0 4 3 0 0 0 4 0 2
0 5 0 1 0 0 0 0 3 2
0 0 0 0 4 4 0 0 0 3
0 0 0 0 4 0 3 0 0 6
0 0 0 0 0 2 2 0 0 1
0 0 0 0 0 0 0 3 6 1 0

V: { 1, 4, 7, 10, 11, 6, 9, 8, 3, 5, 2, }
E:{ (1,4),(4,7),(7,10),(10,11),(10,6),(7,9),(11,8),(6,3),(9,5),(7,2),}

Ваги ребер:
2, 1, 2, 1, 2, 3, 3, 4, 4, 5,

Мінімальне остове дерево: 27

Process returned 0 (0x0)   execution time : 0.115 s
Press any key to continue.
```