

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра систем штучного інтелекту

Лабораторна робота №4

з дисципліни «Дискретна математика»

Виконав:

студент групи КН-112

Тиський Святослав

Викладач:

Мельникова Н.І.

Львів-2019

Варіант №14

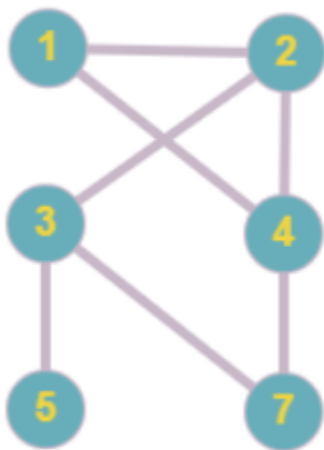
Лабораторна робота № 4.

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

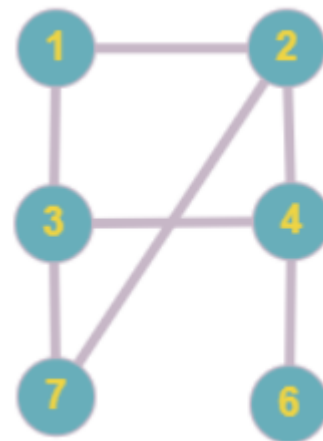
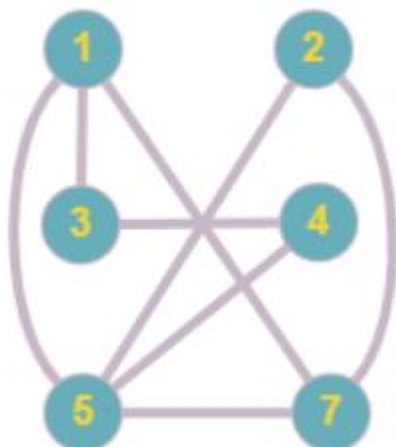
Мета роботи: набуття практичних вмінь та навичок з використання алгоритмів Пріма і Краскала.

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

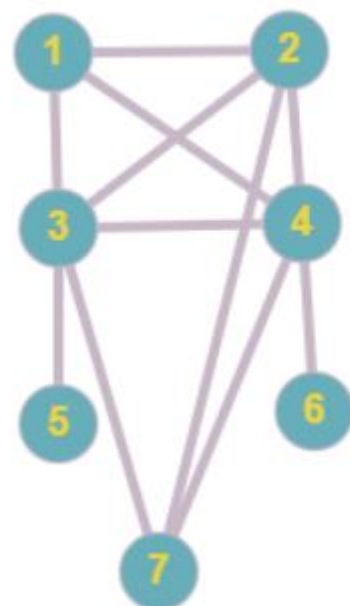
1. Виконати наступні операції над графами: 1) знайти доповнення до першого графу, 2) об'єднання графів, 3) кільцеву суму $G1$ та $G2$ ($G1+G2$), 4) розщепити вершину у другому графі, 5) виділити підграф A , що складається з 3-х вершин в $G1$ і знайти стягнення A в $G1$ ($G1 \setminus A$), 6) добуток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



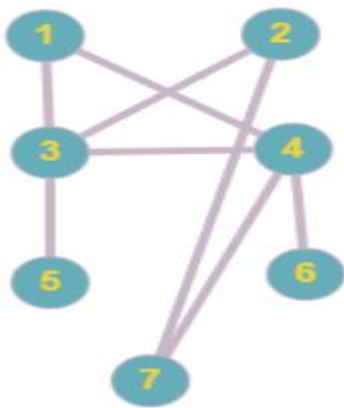
Розв'язок завдання №1.1:



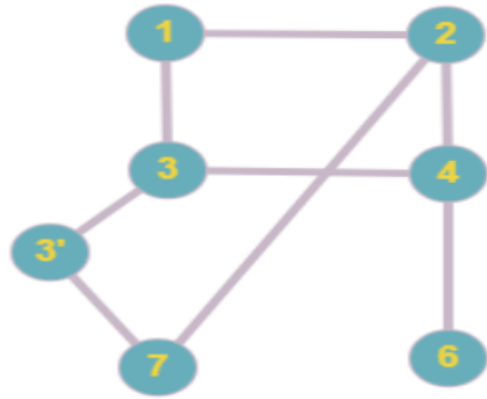
Розв'язок завдання №1.2:



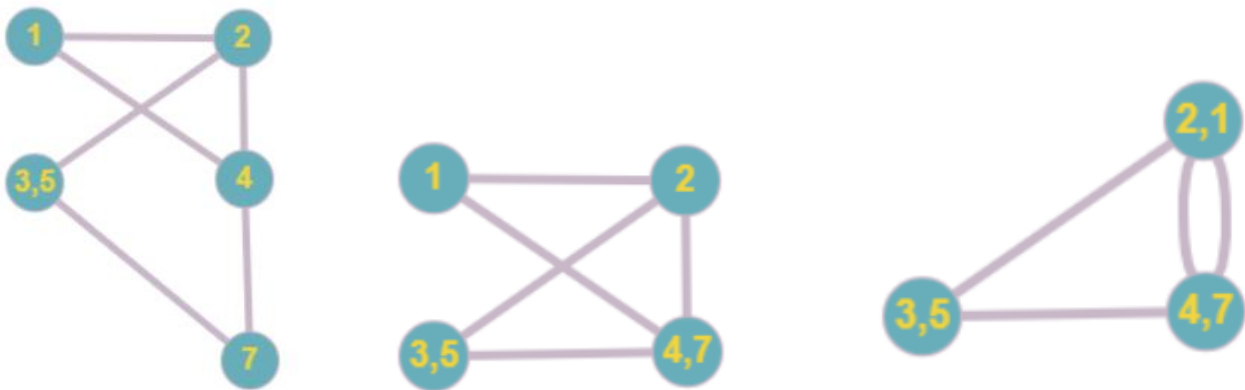
Розв'язок завдання №1.3:



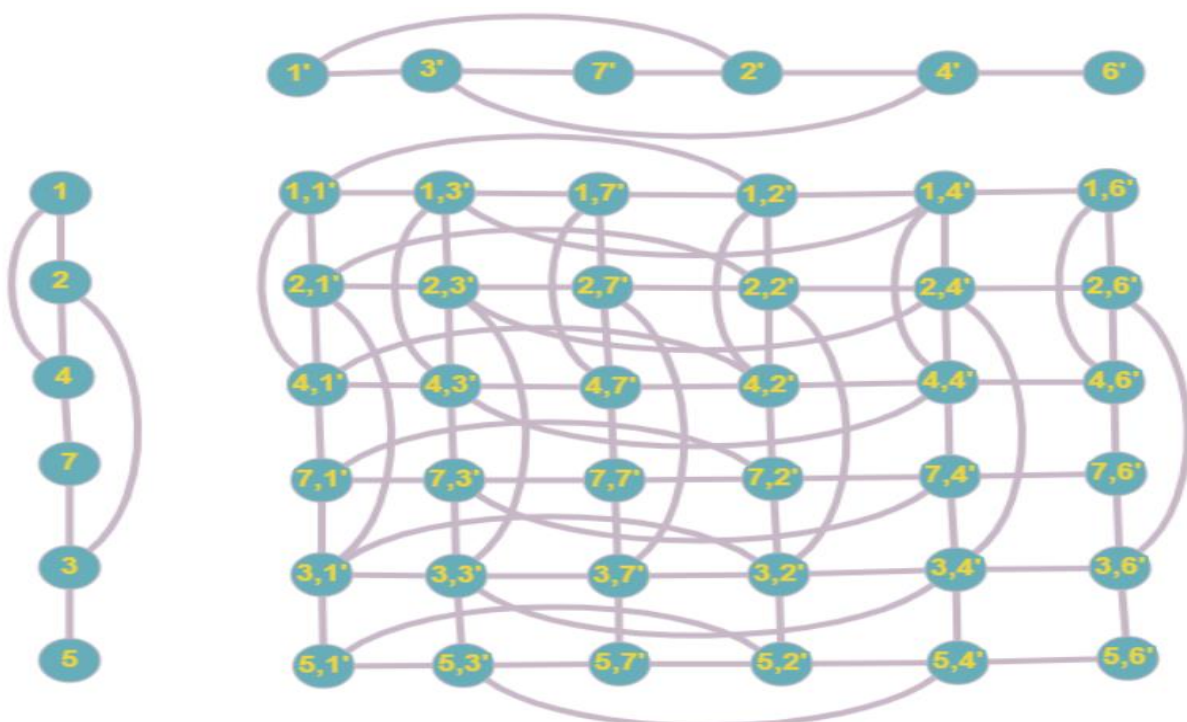
Розв'язок завдання №1.4:



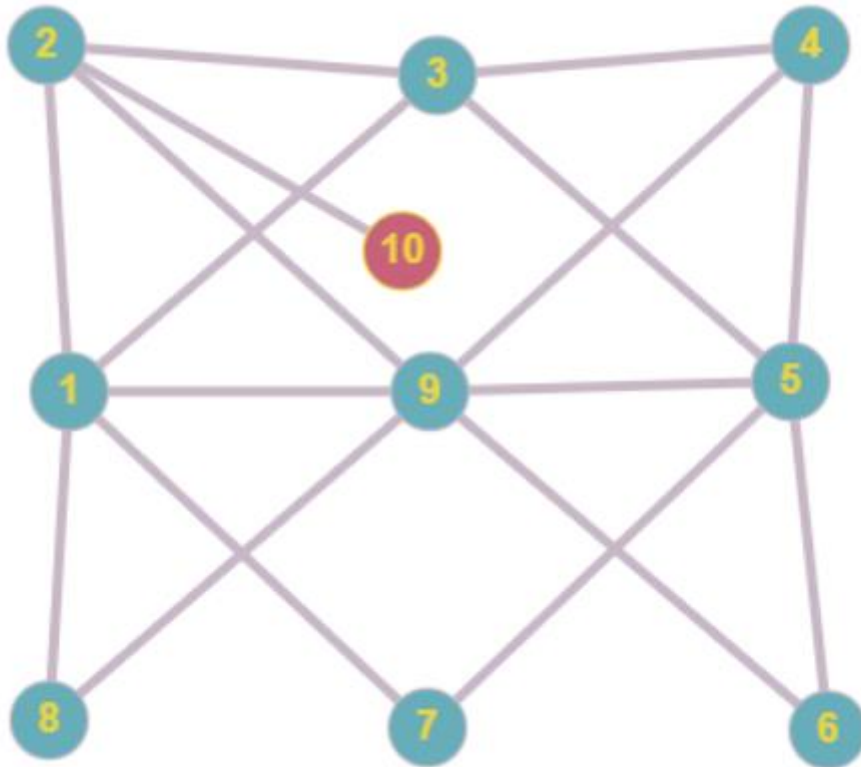
Розв'язок завдання №1.5:



Розв'язок завдання №1.6:



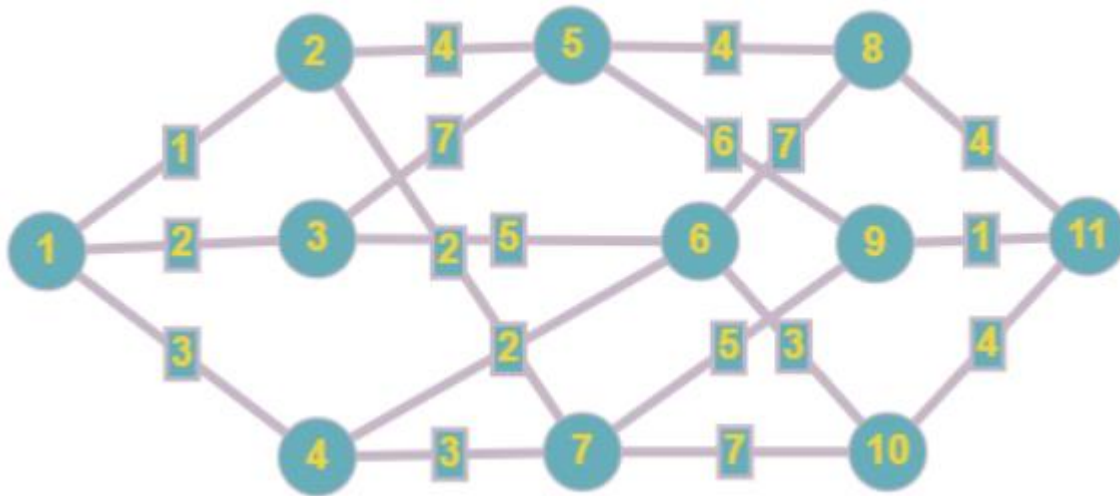
2. Знайти таблицю суміжності та діаметр графа.



Розв'язок завдання №2: Діаметр=3

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----|---|---|---|---|---|---|---|---|---|----|
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 2 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 3 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 5 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 6 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 7 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 8 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 9 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| 10 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

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



Прима:

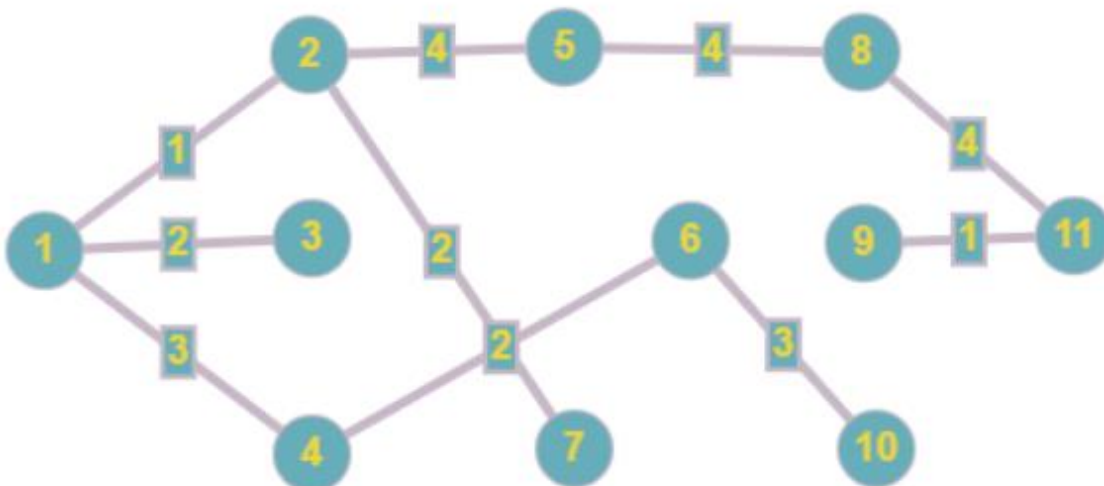
$V: \{1, 2, 3, 7, 4, 6, 10, 5, 8, 11, 9\};$

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

Краскал:

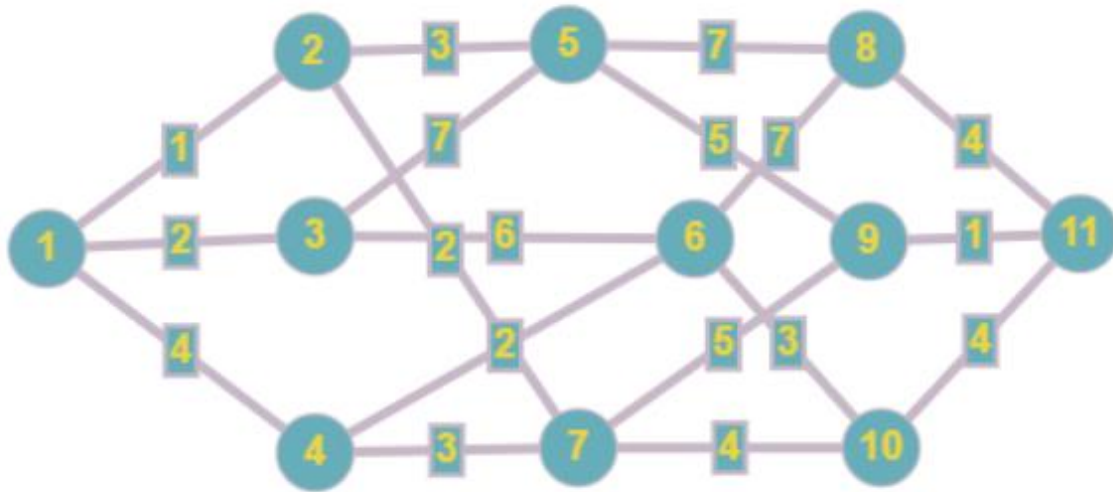
$V: \{1, 2, 9, 11, 3, 7, 4, 6, 10, 5, 8\};$

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



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

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



```
1  #include <iostream>
2
3  using namespace std;
4
5
6
7  int main()
8  {
9      int rebro[18][4] = {
10         {1,2,1,0}, //1ver 2ver masa
11         {1,3,2,0},
12         {1,4,4,0},
13         {2,5,3,0},
14         {2,7,2,0},
15         {3,5,7,0},
16         {3,6,6,0},
17         {4,6,2,0},
18         {4,7,3,0},
19         {5,8,7,0},
20         {5,9,5,0},
21         {6,8,7,0},
22         {6,10,3,0},
23         {7,9,5,0},
24         {7,10,4,0},
25         {8,11,4,0},
26         {9,11,1,0},
27         {10,11,4,0}
28     };
29     int top[11][2] = {
30         {1,0},
31         {2,0},
32         {3,0},
33         {4,0},
34         {5,0},
35         {6,0},
36         {7,0},
37         {8,0},
38         {9,0},
39         {10,0},
40         {11,0}
41     };
42
43     for (int i = 0; i < 18; i++) {
44         for (int k = 0; k < 17-i; k++) {
45             if (rebro[k][2] > rebro[k+1][2]) {
46                 for (int j = 0; j < 4; j++) {
47                     a = rebro[k+1][j];
48                     rebro[k+1][j] = rebro[k][j];
49                     rebro[k][j] = a;
50                 }
51             }
52         }
53     }
```

```
51
52
53     }
54     int count = 1;
55     for (int i = 0; i < 18; i++) {
56         if (top[rebro[i][0] - 1][1] != top[rebro[i][1] - 1][1]) {
57             rebro[i][3] = 1;
58             if (top[rebro[i][0] - 1][1] == 0 && top[rebro[i][1] - 1][1] != 0) {
59                 top[rebro[i][0] - 1][1] = top[rebro[i][1] - 1][1];
60             }
61             if (top[rebro[i][0] - 1][1] != 0 && top[rebro[i][1] - 1][1] == 0) {
62                 top[rebro[i][1] - 1][1] = top[rebro[i][0] - 1][1];
63             }
64             if (top[rebro[i][0] - 1][1] != 0 && top[rebro[i][1] - 1][1] != 0) {
65                 for (int j = 0; j < 11; j++) {
66                     if (top[j][1] == top[rebro[i][1] - 1][1]) {
67                         top[j][1] = top[rebro[i][0] - 1][1];
68                     }
69                 }
70             }
71         }
72     }
73     if (top[rebro[i][0] - 1][1] == 0 && top[rebro[i][1] - 1][1] == 0) {
74         rebro[i][3] = 1;
75         top[rebro[i][0] - 1][1] = count;
76         top[rebro[i][1] - 1][1] = count;
77         count++;
78     }
79 }
80
81 for (int i = 0; i < 18; i++) {
82     if (rebro[i][3] == 1) {
83         cout << rebro[i][0] << " " << rebro[i][1] << " " << rebro[i][2] << endl;
84     }
85 }
86 return 0;
87 }
```

Результат:

| | | |
|----|----|---|
| 1 | 2 | 1 |
| 9 | 11 | 1 |
| 1 | 3 | 2 |
| 2 | 7 | 2 |
| 4 | 6 | 2 |
| 2 | 5 | 3 |
| 4 | 7 | 3 |
| 6 | 10 | 3 |
| 8 | 11 | 4 |
| 10 | 11 | 4 |

Висновок: на цій лабораторній роботі я набув практичних вмінь та навичок з використання алгоритмів Пріма і Краскала.