

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТУ
“ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра систем штучного інтелекту

Лабораторна робота №4

з дисципліни

«Дискретна математика»

Варіант 11

Виконала:

студентка групи КН-112
Подопригора Х.І.

Викладач:

Мельникова Н.І.

Львів – 2019 р.

Тема роботи: 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.

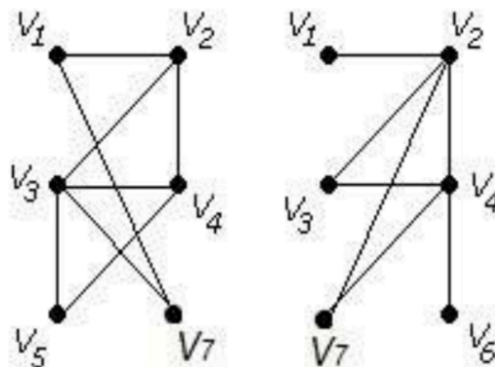
Варіант 11.

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

1. Виконати наступні операції над графами:

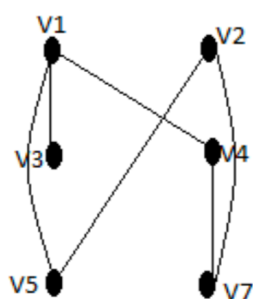
- 1) знайти доповнення до першого графу,
- 2) об'єднання графів,
- 3) кільцеву суму $G1$ та $G2$ ($G1+G2$),
- 4) розщепити вершину у другому графі,
- 5) виділити підграф A , що складається з 3-х вершин в $G1$ і знайти стягнення A в $G1$ ($G1 \setminus A$), 6) добуток графів.

11

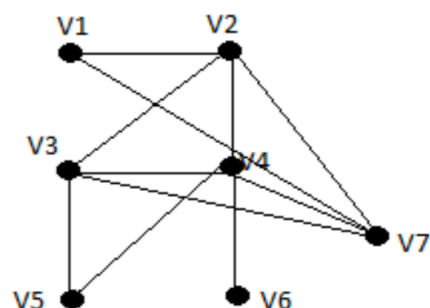


Розв'язок:

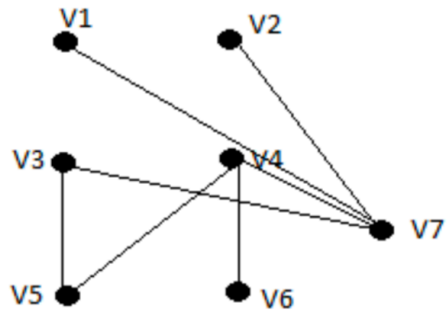
1)



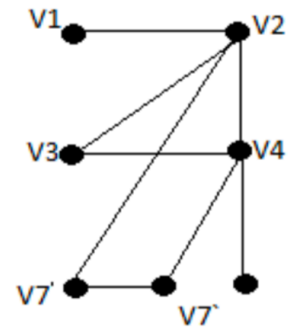
2)



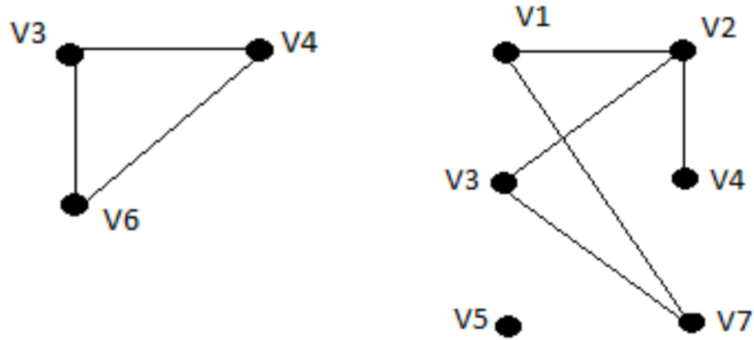
3)



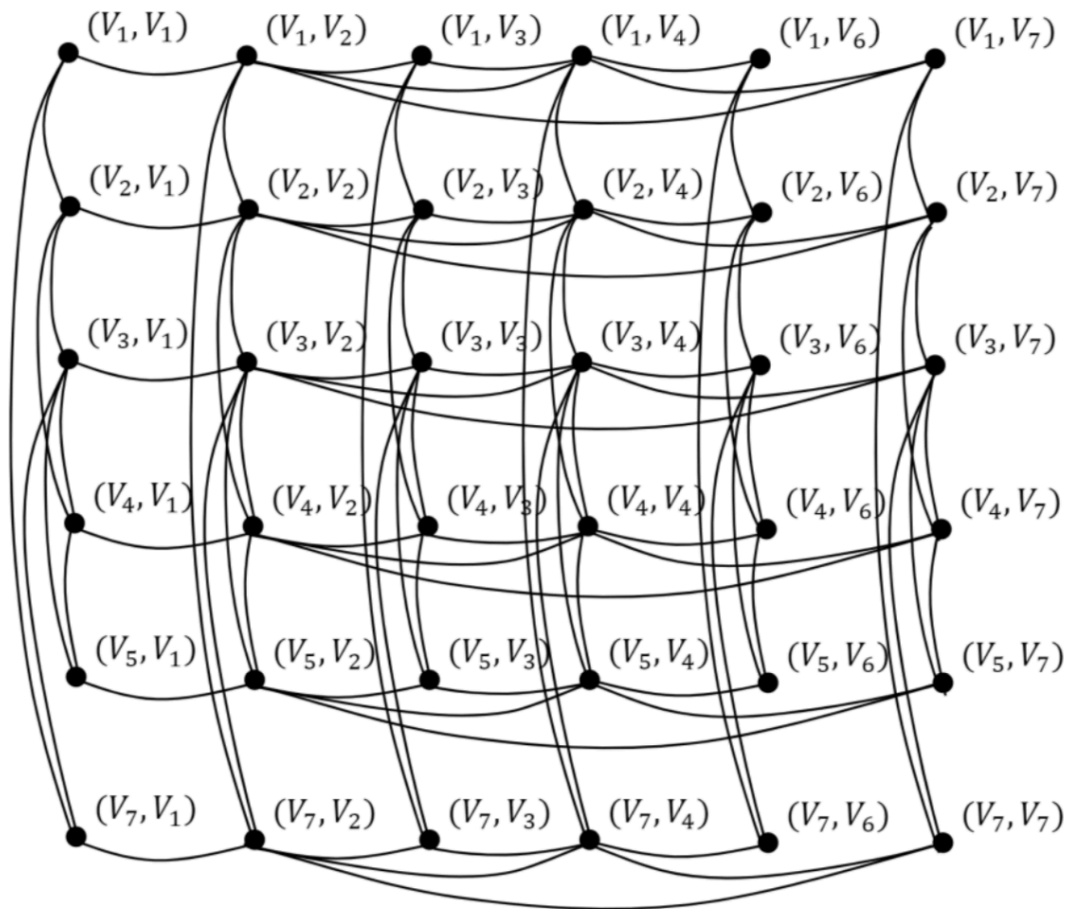
4)



5)

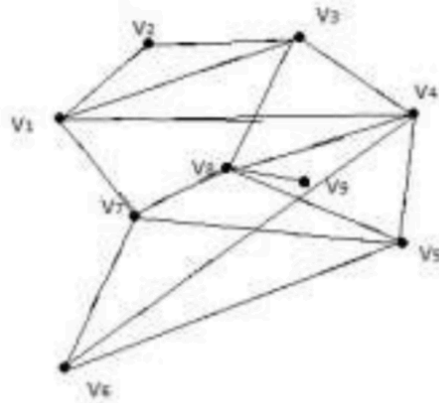


6)



2. Знайти таблицю суміжності та діаметр графа.

11

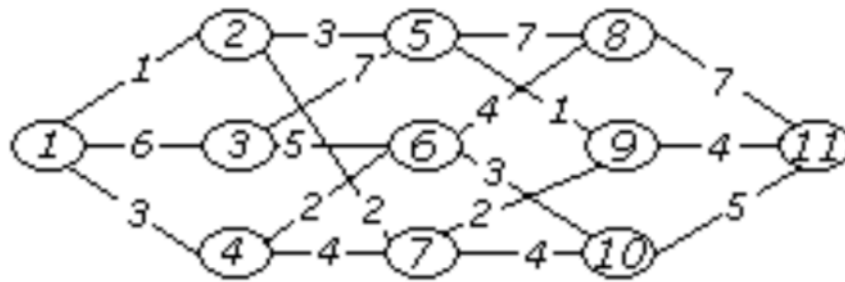


	V1	V2	V3	V4	V5	V6	V7	V8	V9
V1	0	1	1	1	0	0	1	0	0
V2	1	0	1	0	0	0	0	0	0
V3	1	1	0	1	0	0	0	1	0
V4	1	0	1	0	1	1	0	1	0
V5	0	0	0	1	0	1	1	1	0
V6	0	0	0	1	1	0	1	0	0
V7	1	0	0	1	1	1	0	1	0
V8	0	0	1	1	1	0	1	0	1
V9	0	0	0	0	0	0	0	1	0

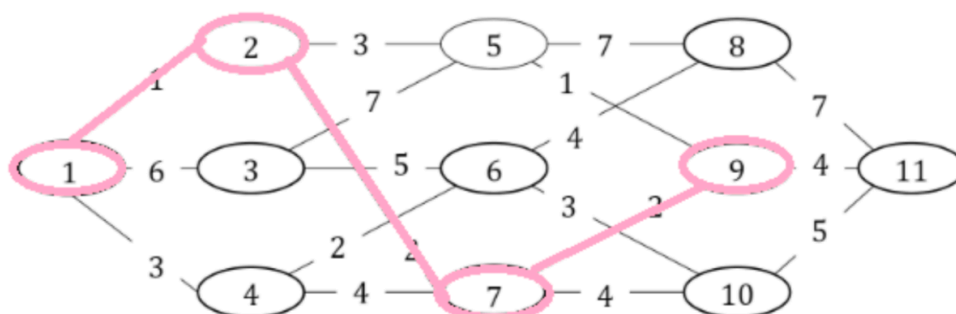
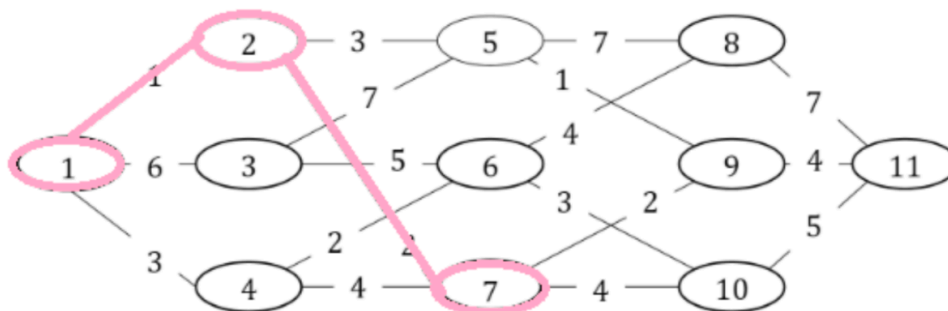
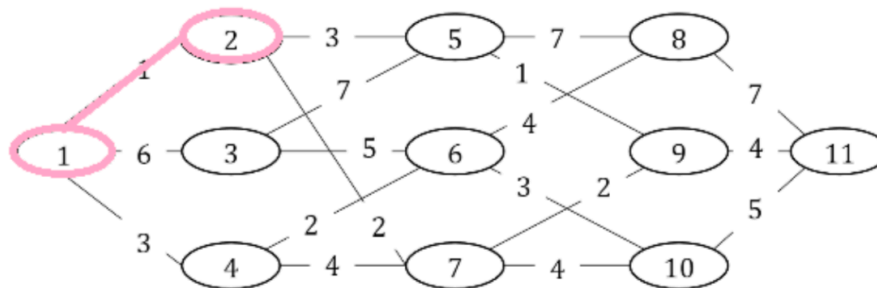
Діаметр графа дорівнює 3, тому що найдовша відстань між двома його вершинами не перевищує 3 (від першої вершини до дев'ятої).

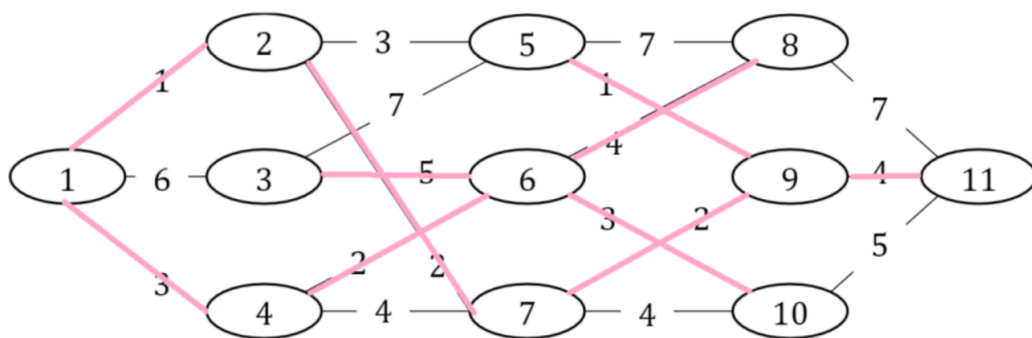
3. Знайти двома методами (Краскала і Прима) мінімальне остове дерево графа.

11

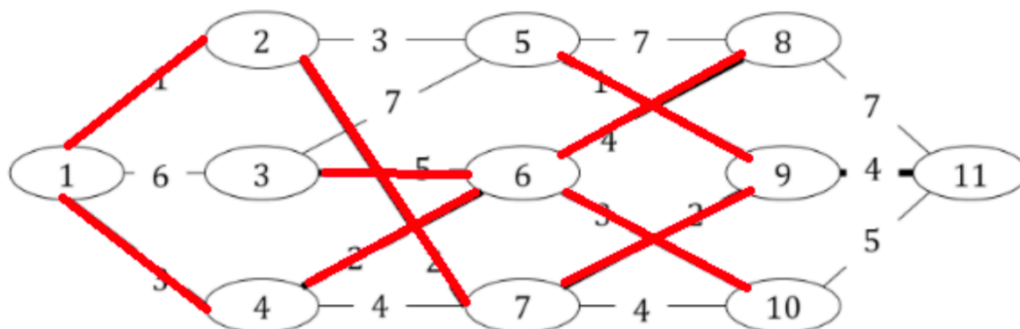
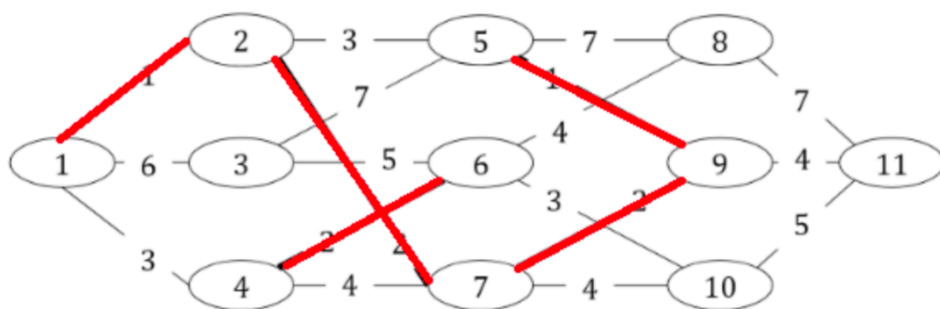
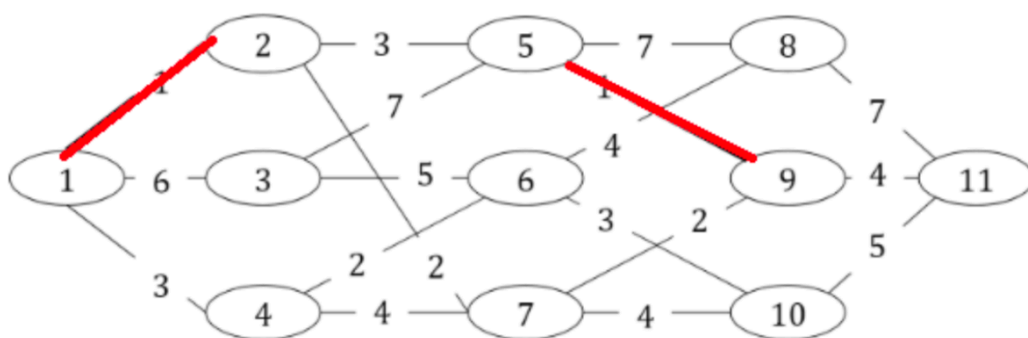


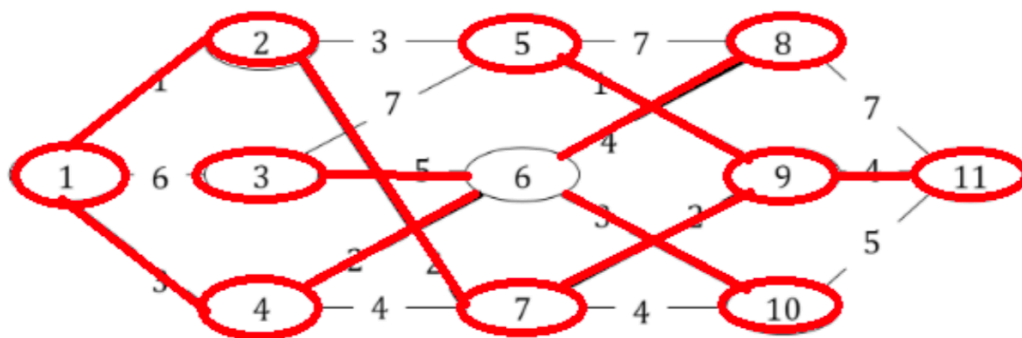
Алгоритм Прима:





Алгоритм Краскала:

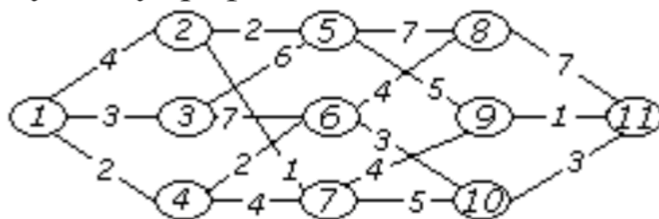




Завдання II.

Варіант № 11

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



Текст програми:

```

9  #include <iostream>
10 using namespace std;
11
12 int main(){
13     int graph[11][11] = {{ 0, 4, 3, 2, 0, 0, 0, 0, 0, 0, 0 },
14                           { 4, 0, 0, 0, 2, 0, 1, 0, 0, 0, 0 },
15                           { 3, 0, 0, 0, 6, 7, 0, 0, 0, 0, 0 },
16                           { 2, 0, 0, 0, 0, 2, 4, 0, 0, 0, 0 },
17                           { 0, 2, 6, 0, 0, 0, 0, 7, 5, 0, 0 },
18                           { 0, 0, 7, 2, 0, 0, 0, 4, 0, 3, 0 },
19                           { 0, 1, 0, 4, 0, 0, 0, 0, 4, 5, 0 },
20                           { 0, 0, 0, 0, 7, 4, 0, 0, 0, 0, 7 },
21                           { 0, 0, 0, 0, 5, 0, 4, 0, 0, 0, 1 },
22                           { 0, 0, 0, 0, 0, 3, 5, 0, 0, 0, 3 },
23                           { 0, 0, 0, 0, 0, 0, 0, 7, 1, 3, 0 } };
24
25     int n=11;
26     int sumWeight = 0;

```

```

27     bool *visited = new bool[n];
28     memset(visited, false, sizeof(bool)*n);
29     visited[0] = true;
30     for(int l=0; l<n-1; l++){
31         int minx = -1 , miny = -1 ;
32         for(int i=0; i<n; i++)
33             if(visited[i])
34                 for(int j=0; j<n; j++)
35                     if(!visited[j] && graph[i][j] > 0 &&
36                         (miny == -1 || graph[i][j] < graph[miny][minx]))
37                         static_cast<void>(miny = i), minx = j;
38         visited[minx] = true;
39         sumWeight += graph[miny][minx];
40         cout << miny+1 << " -- " << minx+1 << " weight: " << graph[miny][minx] << endl;
41     }
42     cout << "Total weight of our new tree: " << sumWeight << endl;
43
44     return 0;
45 }

```

Результати виконання програми:

```

1 -- 4 weight: 2
4 -- 6 weight: 2
1 -- 3 weight: 3
6 -- 10 weight: 3
10 -- 11 weight: 3
11 -- 9 weight: 1
1 -- 2 weight: 4
2 -- 7 weight: 1
2 -- 5 weight: 2
6 -- 8 weight: 4
Total weight of our new tree: 25
Program ended with exit code: 0

```

Висновок: після виконання лабораторної роботи я навчилася виконувати операції над графами та використовувувати алгоритми Прима і Крaskала для пошуку мінімального остового дерева в графі.