

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТУ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра систем штучного інтелекту

Лабораторна робота № 4

З дисципліни «Дискретна математика»

Виконав:

студент групи КН-112

Матвіїв Остап-Василь

Викладач:

Мельникова Н.І.

Львів – 2019 р.

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.

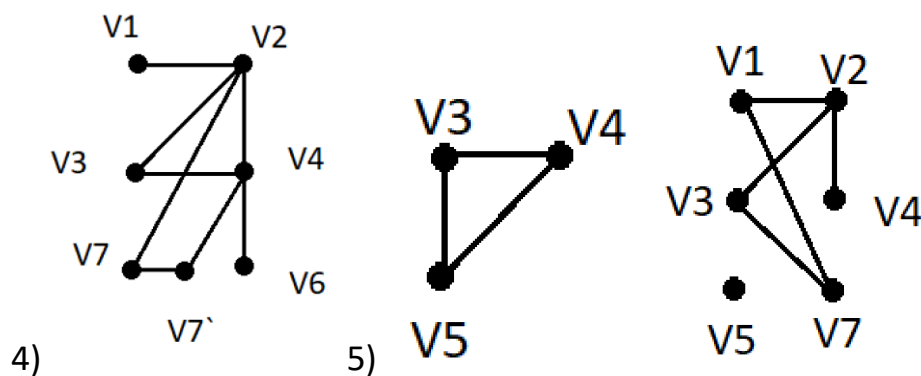
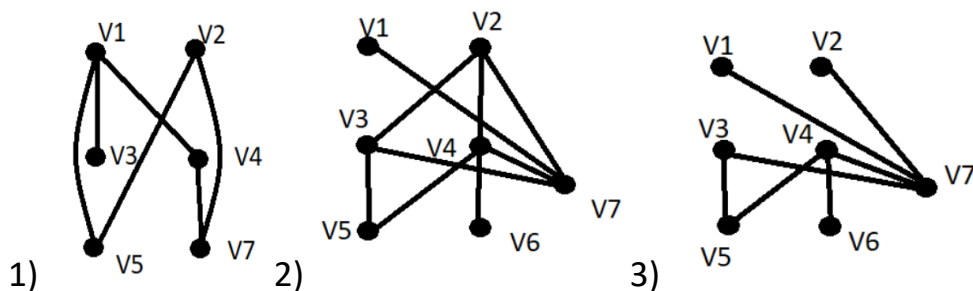
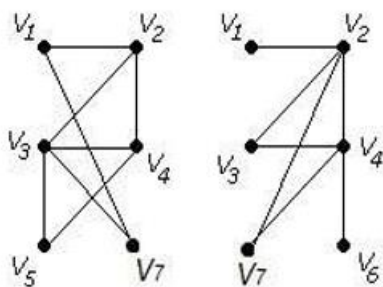
Варіант-11

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.

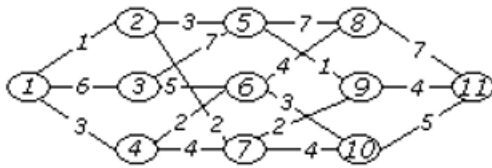
11



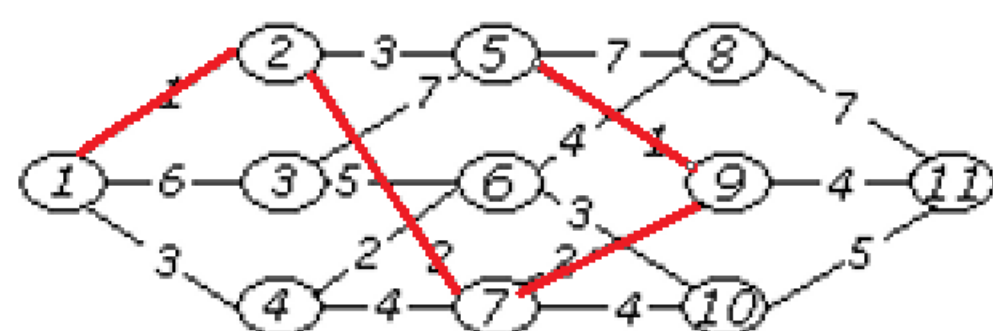
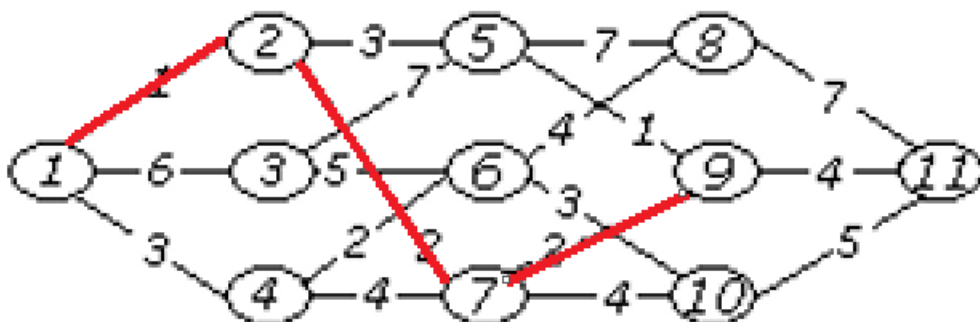
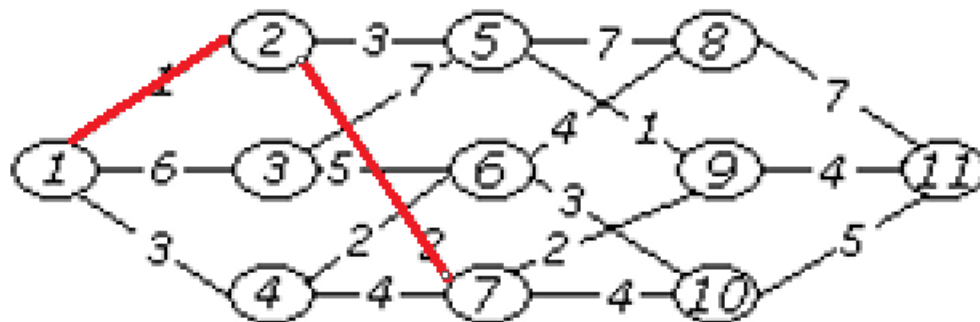
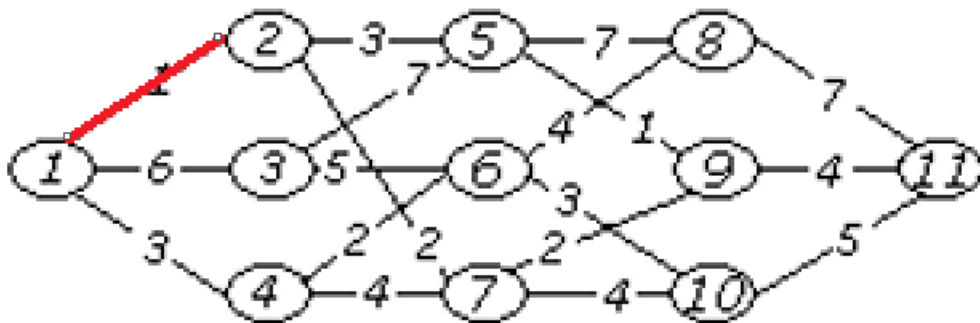
Діаметр графа дорівнює 3, тому що найдовша відстань між двома його вершинами не перевищує 3.

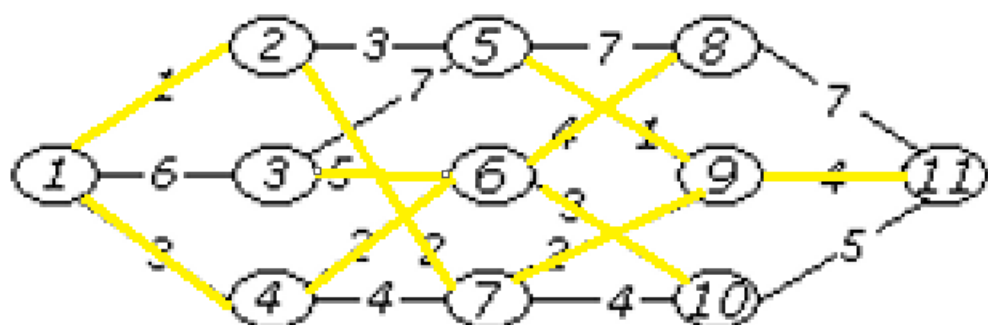
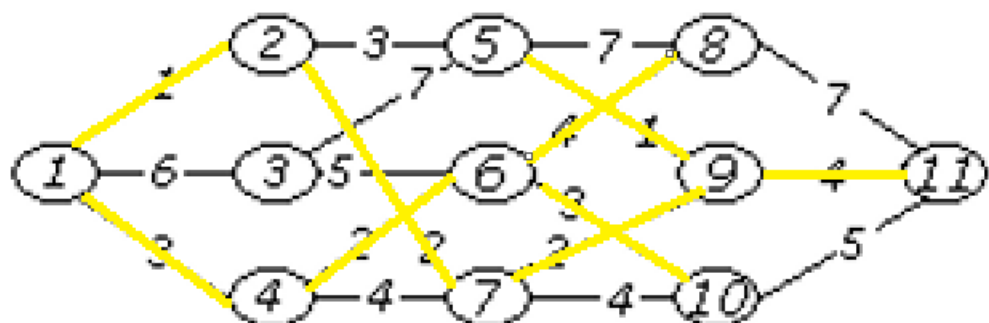
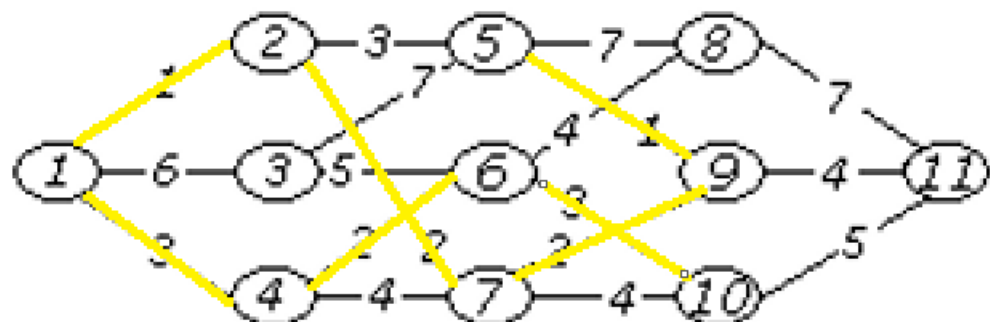
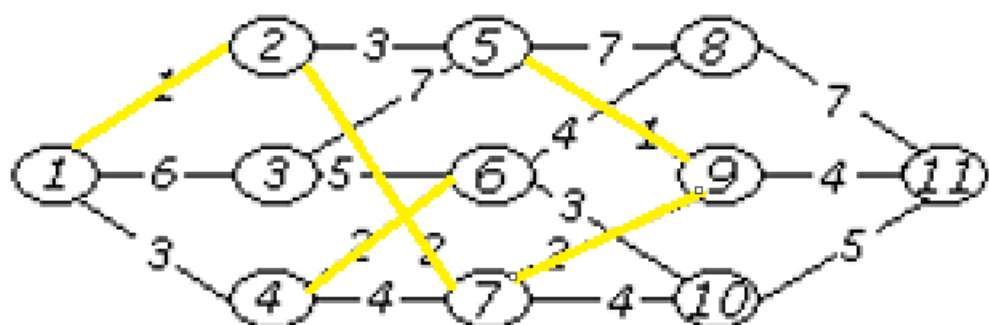
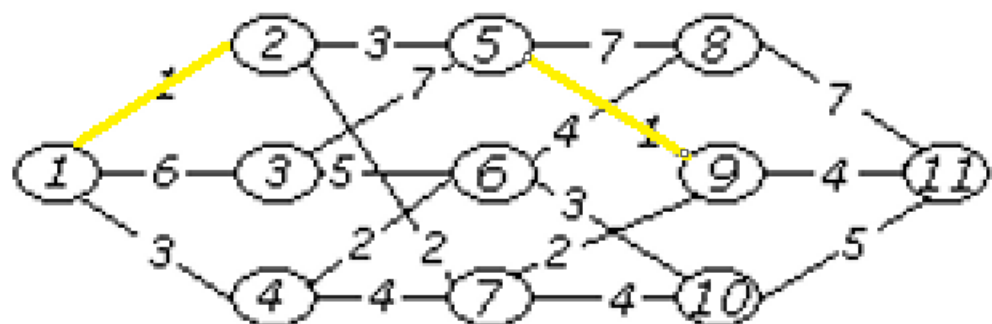
3. Знайти двома методами (Краскала і Прима) мінімальне остове дерево графа.

11



1. Алгоритм Прима:

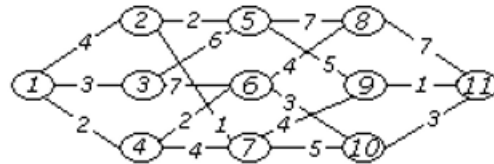




Завдання №2. Написати програму, яка реалізує алгоритм знаходження остового дерева мінімальної ваги згідно свого варіанту.

Варіант № 11

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



```
using namespace std;
int main()
{
    int graph[11][11] = {
        {0,4,3,2,0,0,0,0,0,0,0},
        {4,0,0,0,2,0,1,0,0,0,0},
        {3,0,0,0,6,7,0,0,0,0,0},
        {2,0,0,0,0,2,4,0,0,0,0},
        {0,2,6,0,0,0,0,7,5,0,0},
        {0,0,7,2,0,0,0,4,0,3,0},
        {0,1,0,4,0,0,0,0,4,5,0},
        {0,0,0,0,7,4,0,0,0,0,7},
        {0,0,0,0,5,0,4,0,0,0,1},
        {0,0,0,0,0,3,5,0,0,0,3},
        {0,0,0,0,0,0,0,7,1,3,0}};

    int n = 11, sumweight = 0;
    bool* visited = new bool[n];
    memset(visited, false, sizeof(bool) * n);
    visited[0] = true;
    for (int l = 0; l < n - 1; l++) {
        int minx = -1, miny = -1;
        for (int i = 0; i < n; i++) {
            if (visited[i]) {
                for (int j = 0; j < n; j++) {
                    if (!visited[j] && graph[i][j] > 0 && (minx == -1 || graph[i][j] < graph[minx][miny])) {
                        static_cast<void*>(miny = i), minx = j;
                    }
                }
            }
        }
        visited[minx] = true;
        sumweight += graph[miny][minx];
        cout << miny + 1 << "-" << minx + 1 << " weight: " << graph[miny][minx] << endl;
    }
    cout << "Total weight: " << sumweight << endl;
    return 0;
}
```

```
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: 25
```

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