

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
“ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра систем штучного інтелекту

Лабораторна робота №4
з дисципліни «Дискретна математика»

Виконав:

Студент групи КН-112

Шклярів Віталій

Викладач:

Мельникова Н. І.

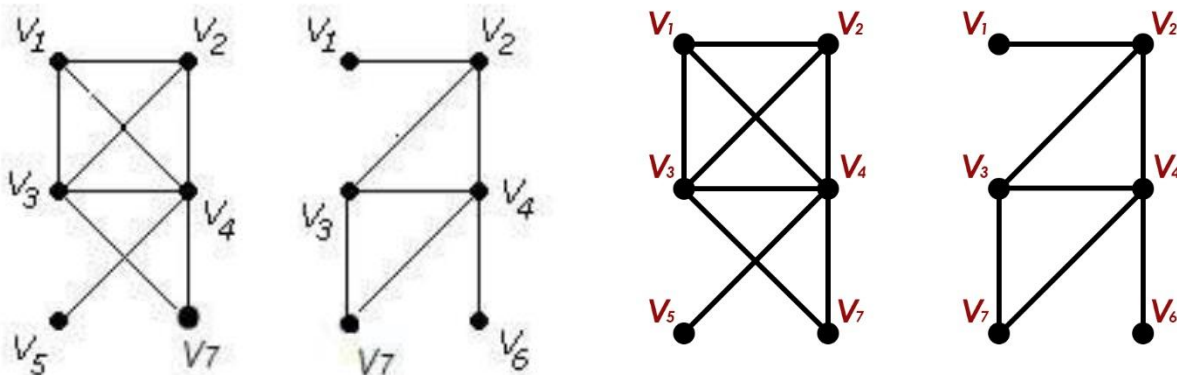
Тема: Основні операції над графами. Знаходження остова мінімальної ваги за алгоритмом Пріма-Краскала.

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

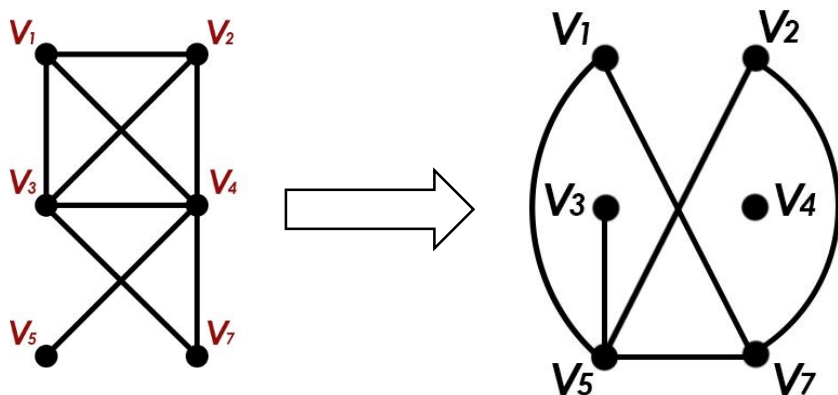
Варіант 15

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

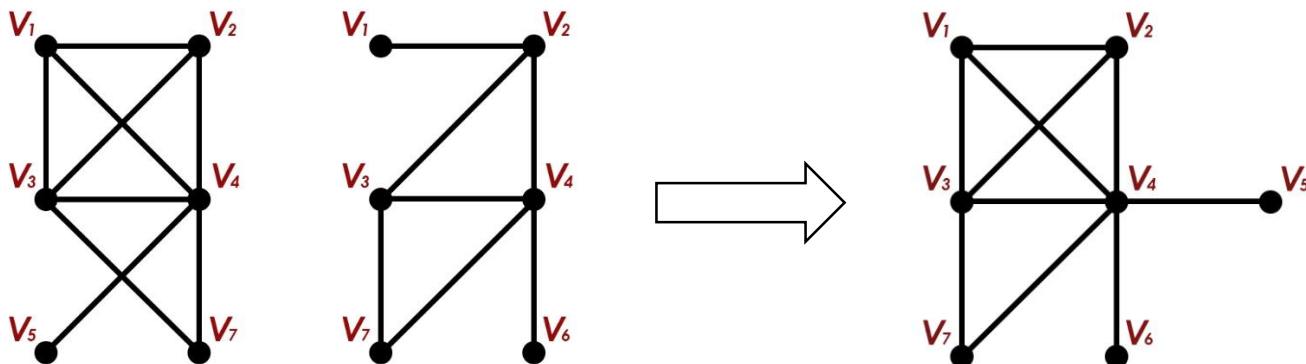
1. Виконати наступні операції над графами:



1) Знайти доповнення до першого графу



2) Об'єднання графів



3) Кільцеву суму $G1$ та $G2$ ($G1+G2$)

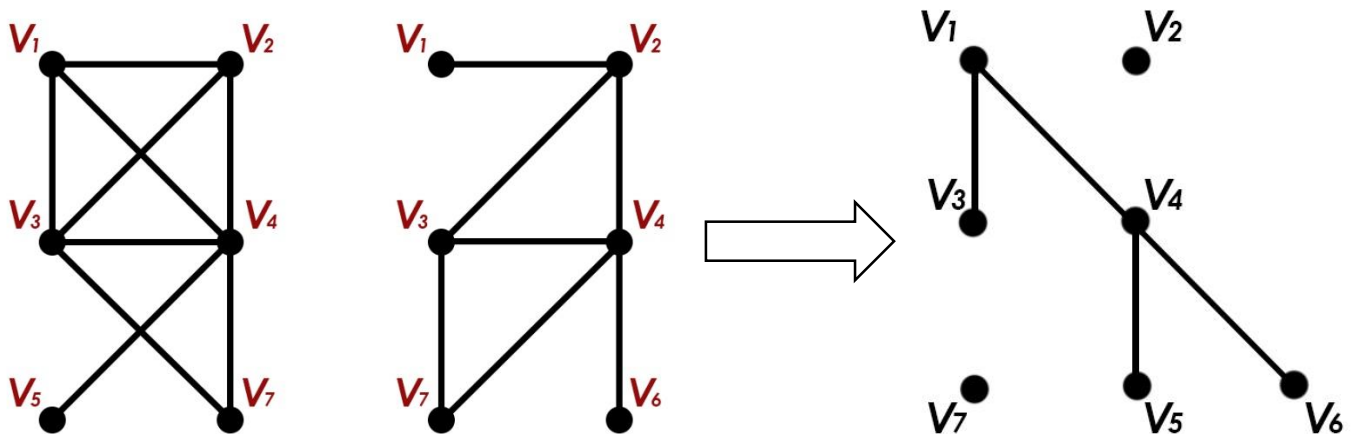
$$V = \{V_1; V_2; V_3; V_4; V_5; V_6; V_7\}$$

$$E = E_{G1} \Delta E_{G2}$$

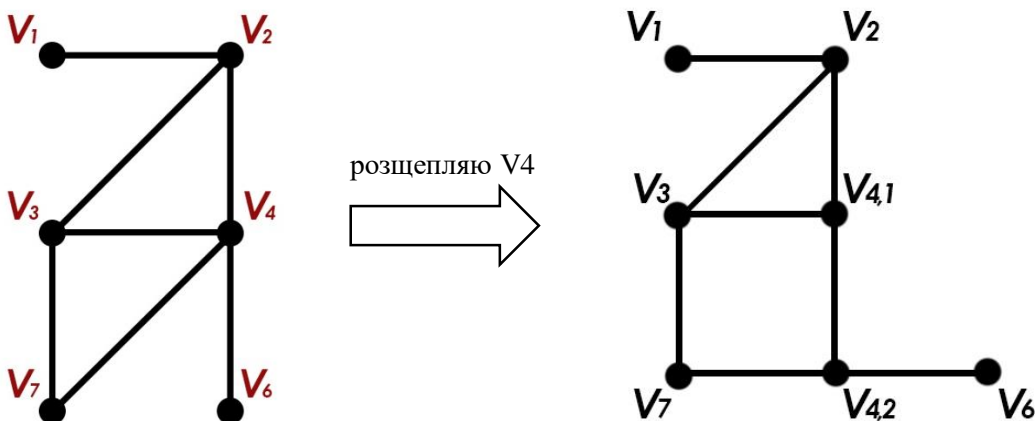
$$E_{G1} = \{(1,2); (1,3); (1,4); (2,3); (2,4); (3,4); (3,7); (4,5); (4,7)\};$$

$$E_{G2} = \{(1,2); (2,3); (2,4); (3,4); (3,7); (4,7); (4,6)\};$$

$$E = \{(1,3); (1,4); (4,5); (4,6)\}$$

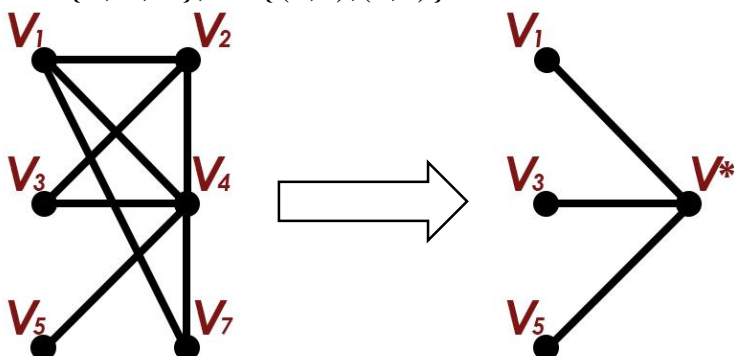


4) Розщепити вершину у другому графі

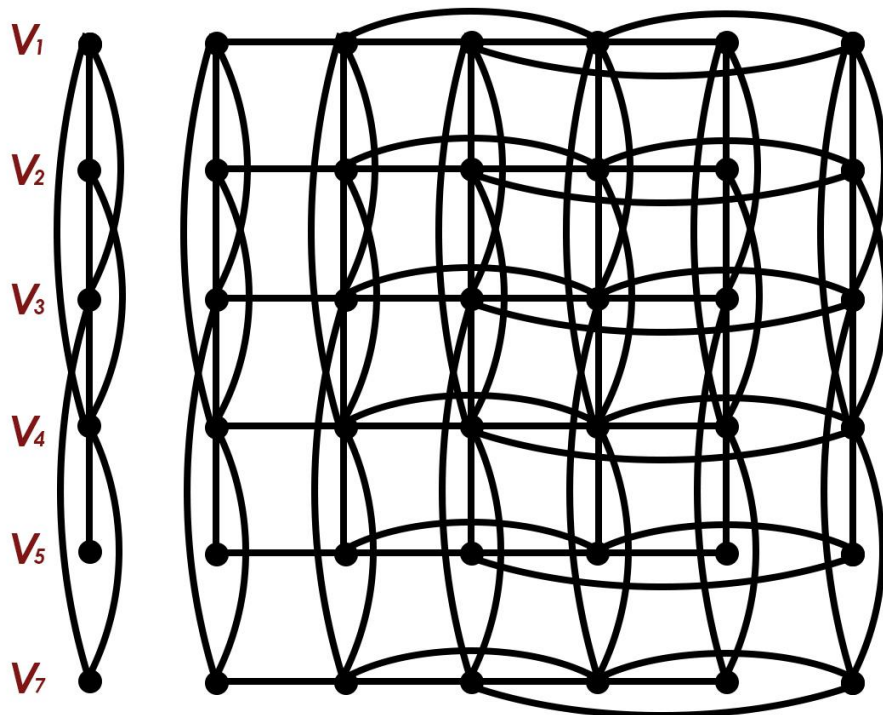
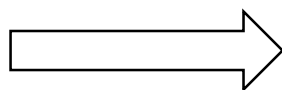
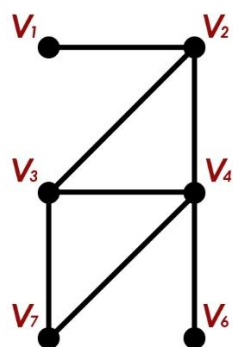
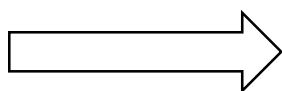
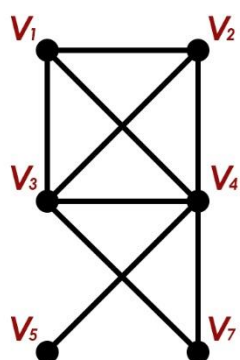


5) Виділити підграф A , що складається з 3-х вершин в $G1$ і знайти стягнення A в $G1$ ($G1 \setminus A$).

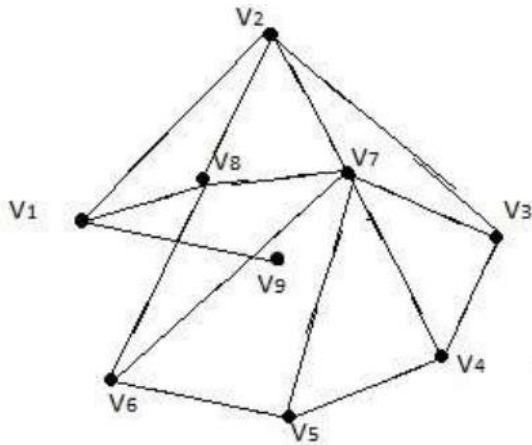
$$A: V = \{2, 4, 7\}, E = \{(2,4); (4,7)\}$$



6) Добуток графів



2. Знайти таблицю суміжності та діаметр графа.



Таблиця суміжності

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

Таблиця відстаней між вершинами ($d = 4$)

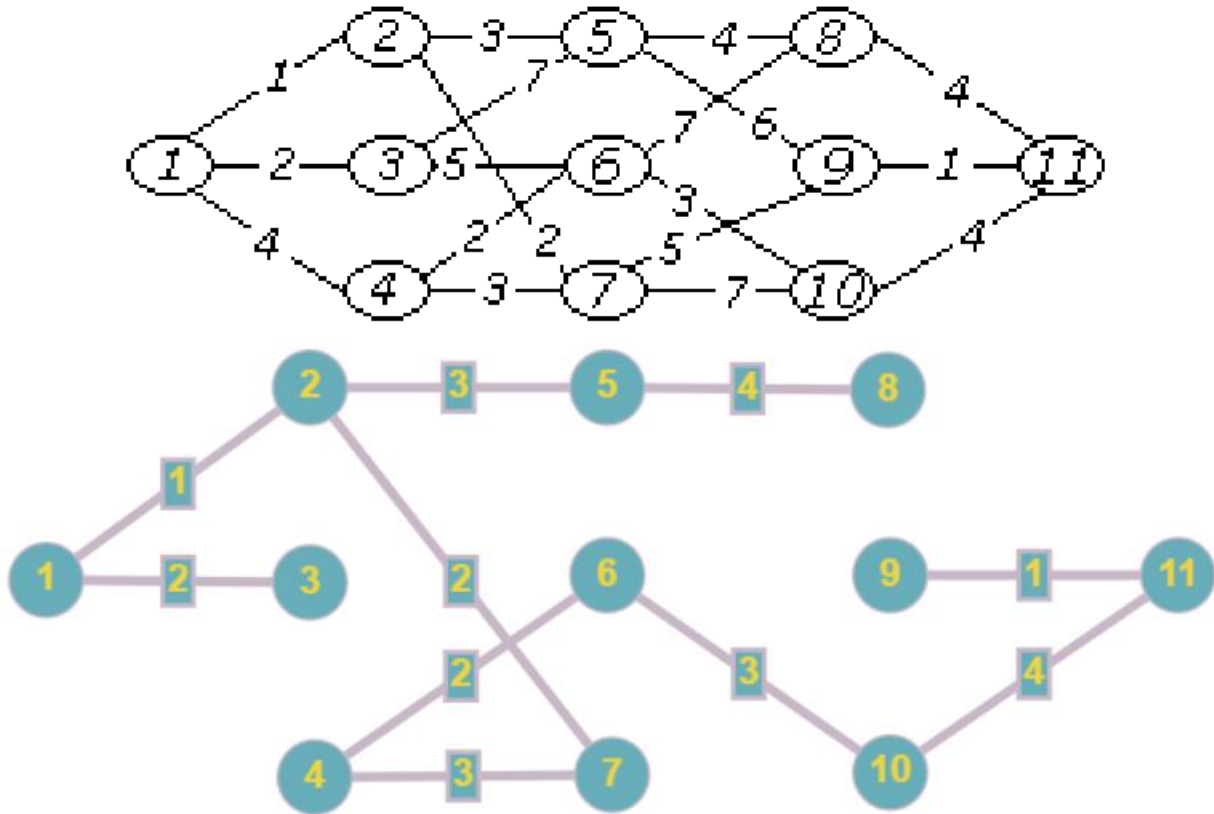
	V1	V2	V3	V4	V5	V6	V7	V8	V9
V1	0	1	2	3	3	2	2	1	1
V2	1	0	1	2	2	2	1	1	2
V3	2	1	0	1	2	2	1	2	3
V4	3	2	1	0	1	2	1	2	4
V5	3	2	2	1	0	1	1	2	4
V6	2	2	2	2	1	0	1	1	3
V7	2	1	1	1	1	1	0	1	3
V8	1	1	2	2	2	1	1	0	2
V9	1	2	3	4	4	3	3	2	0

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

Метод Краскала

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

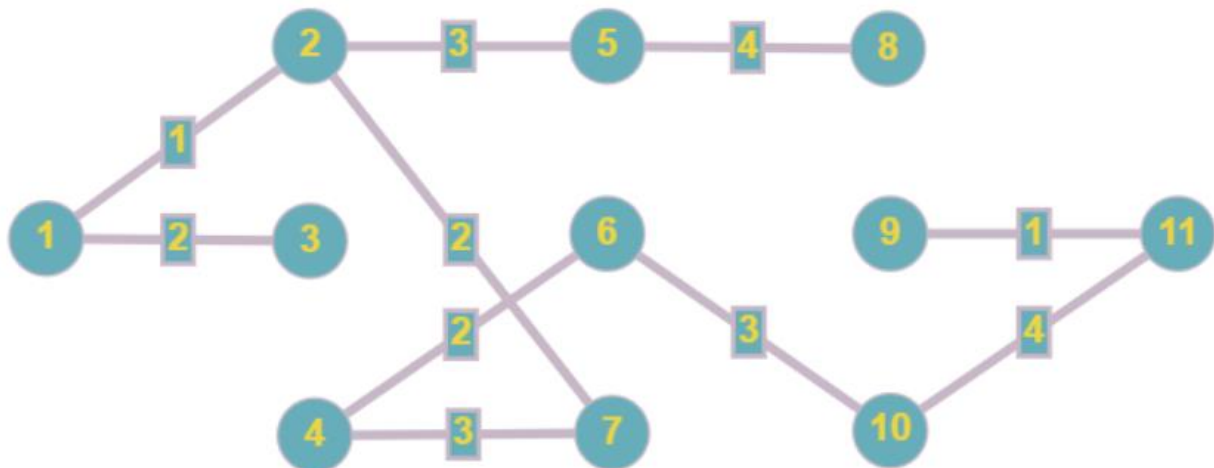
$E : \{(1,2); (9,11); (1,3); (4,6); (2,7); (4,7); (2,5); (6,10); (5,8); (10,11)\}$



Метод Прима

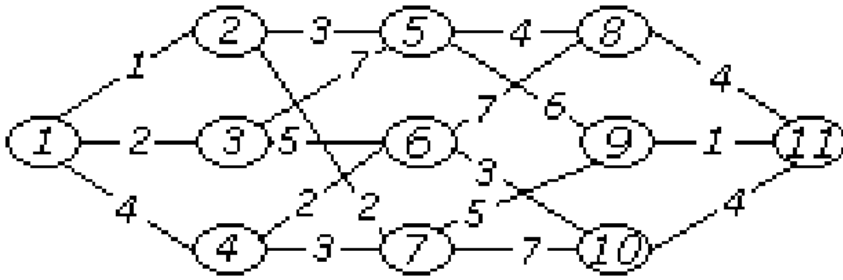
$V : \{1, 2, 3, 7, 5, 4, 6, 10, 8, 11, 9\}$

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



Завдання 2

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



```
#include <iostream>

using namespace std;

int main() {
    int v, count = 0, min = 0, k, t;
    bool check = false;
    cout << "Enter quantity of vertex : ";
    cin >> v;
    int *tops = new int[v];
    int **g = new int *[v];
    int **r = new int *[v - 1];

    for (int j = 0; j < v; j++) {
        g[j] = new int[v];
    }
    for (int j = 0; j < v - 1; j++) {
        r[j] = new int[2];
    }
    for (int a = 0; a < v; a++) {
        for (int j = 0; j < v; j++) {
            cin >> g[a][j];
        }
    }
    tops[count] = 1;
    count++;
    for (int i = 0; count < v; i++) {
        for (int j = 0; j < count; j++) {
            for (int a = 0; a < v; a++) {
                for (int m = 0; m < count; m++) {
                    if (tops[m] == a + 1) {
                        check = true;
                    }
                }
                if (check) {
                    check = false;
                    continue;
                }
                if (min == 0 && g[tops[j] - 1][a] > 0) {
                    min = g[tops[j] - 1][a];
                    k = r[count - 1][0] = tops[j];
                    t = r[count - 1][1] = a + 1;
                    continue;
                }
                if (g[tops[j] - 1][a] > 0 && g[tops[j] - 1][a] < min) {
                    min = g[tops[j] - 1][a];
                    k = r[count - 1][0] = tops[j];
                    t = r[count - 1][1] = a + 1;
                }
            }
        }
        g[k - 1][t - 1] = 0;
        g[t - 1][k - 1] = 0;

        tops[count] = t;
        count++;
        min = 0;
    }
}
```

```
cout << "V: { ";
for (int j = 0; j < v; j++) {
    cout << tops[j] << ", ";
}
cout << "}";
cout << endl << "E:{ ";
for (int j = 0; j < v - 1; j++) {
    cout << "( " << r[j][0] << ", " << r[j][1] << " ), ";
}
cout << "}";

delete [] tops;
for (int i = 0; i < v; i++)
    delete [] g[i];
for (int i = 0; i < v; i++)
    delete [] r[i];

return 0;
}
```

Вивід програми:

```
Enter quantity of vertex :11
0 1 2 4 0 0 0 0 0 0 0
1 0 0 0 3 0 2 0 0 0 0
2 0 0 0 7 5 0 0 0 0 0
4 0 0 0 0 2 3 0 0 0 0
0 3 7 0 0 0 0 4 6 0 0
0 0 5 2 0 0 0 7 0 3 0
0 2 0 3 0 0 0 0 5 7 0
0 0 0 0 4 7 0 0 0 0 4
0 0 0 0 6 0 5 0 0 0 1
0 0 0 0 0 3 7 0 0 0 4
0 0 0 0 0 0 0 4 1 4 0

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

Висновок:

На цій лабораторній роботі я набув практичних вмінь та навичок з використання алгоритмів Пріма і Краскала. Виконувати основні операції над графами.