

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА
ПОЛІТЕХНІКА»**

Інститут комп'ютерних наук та інформаційних технологій

Кафедра систем штучного інтелекту



Лабораторна робота №3

з дисципліни

“Дискретна математика ”

Виконала:

ст. гр. КН-110

Кручковська Христина

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.

ТЕОРЕТИЧНІ ВІДОМОСТІ ТА ПРИКЛАДИ РОЗВ'ЯЗАННЯ ЗАДАЧ

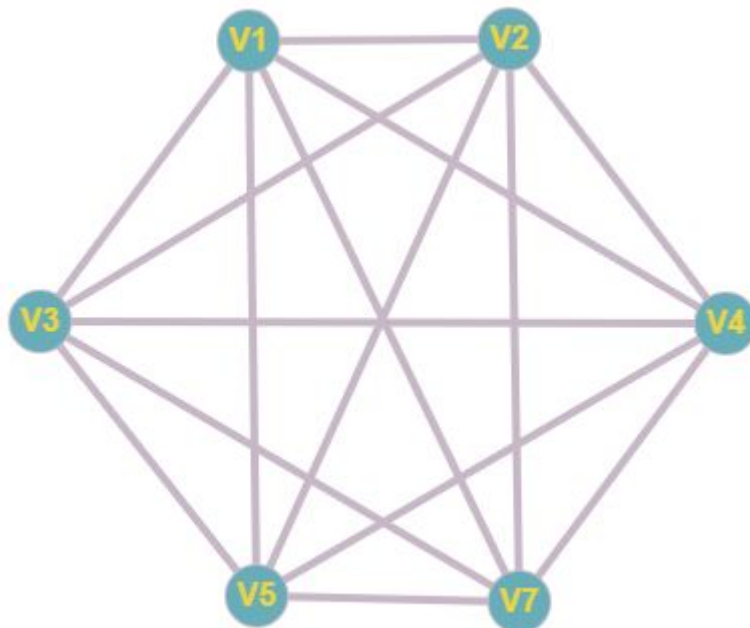
Теорія графів дає простий, доступний і потужний інструмент побудови моделей прикладних задач, є ефективним засобом формалізації сучасних інженерних і наукових задач у різних областях знань. Графом G називається пара множин V, E , де V – множина вершин, перенумерованих числами $1, 2, \dots, n$; V, E – множина упорядкованих або неупорядкованих пар $e = (v', v'')$, $v' \in V$, $v'' \in V$, називаних дугами або ребрами, $E = \{e\}$. При цьому не має примусового значення, як вершини розташовані в просторі або площині і які конфігурації мають ребра. Неорієнтованим графом G називається граф у якого ребра не мають напрямку. Такі ребра описуються неупорядкованою парою (v', v'') . Орієнтований граф (орграф) – це граф ребра якого мають напрямок та можуть бути описані упорядкованою парою (v', v'') . Упорядковане ребро називають дугою. Граф є змішаним, якщо наряду з орієнтованими ребрами (дугами) є також і неорієнтовані. При розв'язку задач змішаний граф зводиться до орграфа. Кратними (паралельними) називаються ребра, які зв'язують одні і ті ж вершини. Якщо ребро виходить та й входить у дну і ту саму вершину, то таке ребро називається петлею. Мультиграф – граф, який має кратні ребра. Псевдограф – граф, який має петлі. Простий граф – граф, який не має кратних ребер та петель. Будь яке ребро є інцедентно двом вершинам (v', v'') , які воно з'єднує. У свою чергу вершини (v', v'') інцедентні до ребра e . Дві вершини (v', v'') називають суміжними, якщо вони належать до одного й того самого ребра e , і несуміжні у протилежному випадку. Два ребра називають суміжними, якщо вони мають спільну вершину. Відношення суміжності як для вершин, так і

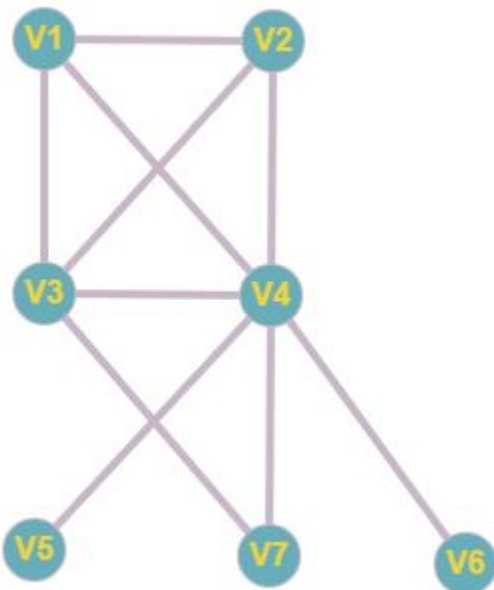
для ребер є симетричним відношенням. Степенем вершини графа G називається число інцидентних їй ребер. Граф, який не має ребер називається пустим графом, нульграфом. Вершина графа, яка не інцидентна до жодного ребра, називається ізольованою. Вершина графа, яка інцидентна тільки до одного ребра, називається звисаючою. Частина $G[V', E']$ графа $G[V, E]$ називається підграфом графа G , якщо $V' \subseteq V$ і E' складається з тих і тільки тих ребер $e = (v', v'')$, у яких обидві кінцеві вершини $v', v'' \in V'$. Частина $G[V', E]$ називається суграфом або остовим підграфом графа G , якщо виконано умови: $V' = V, E' \subseteq E$.

Варіант №15

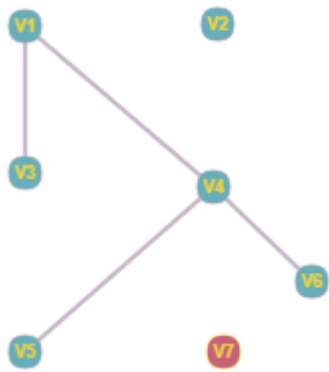
Завдання №1

1. 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.

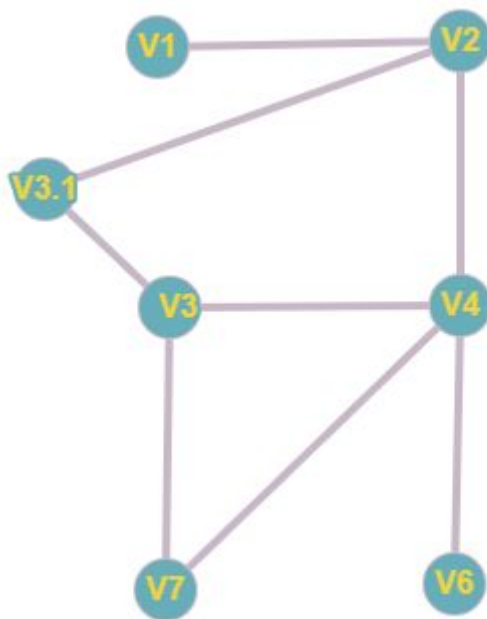




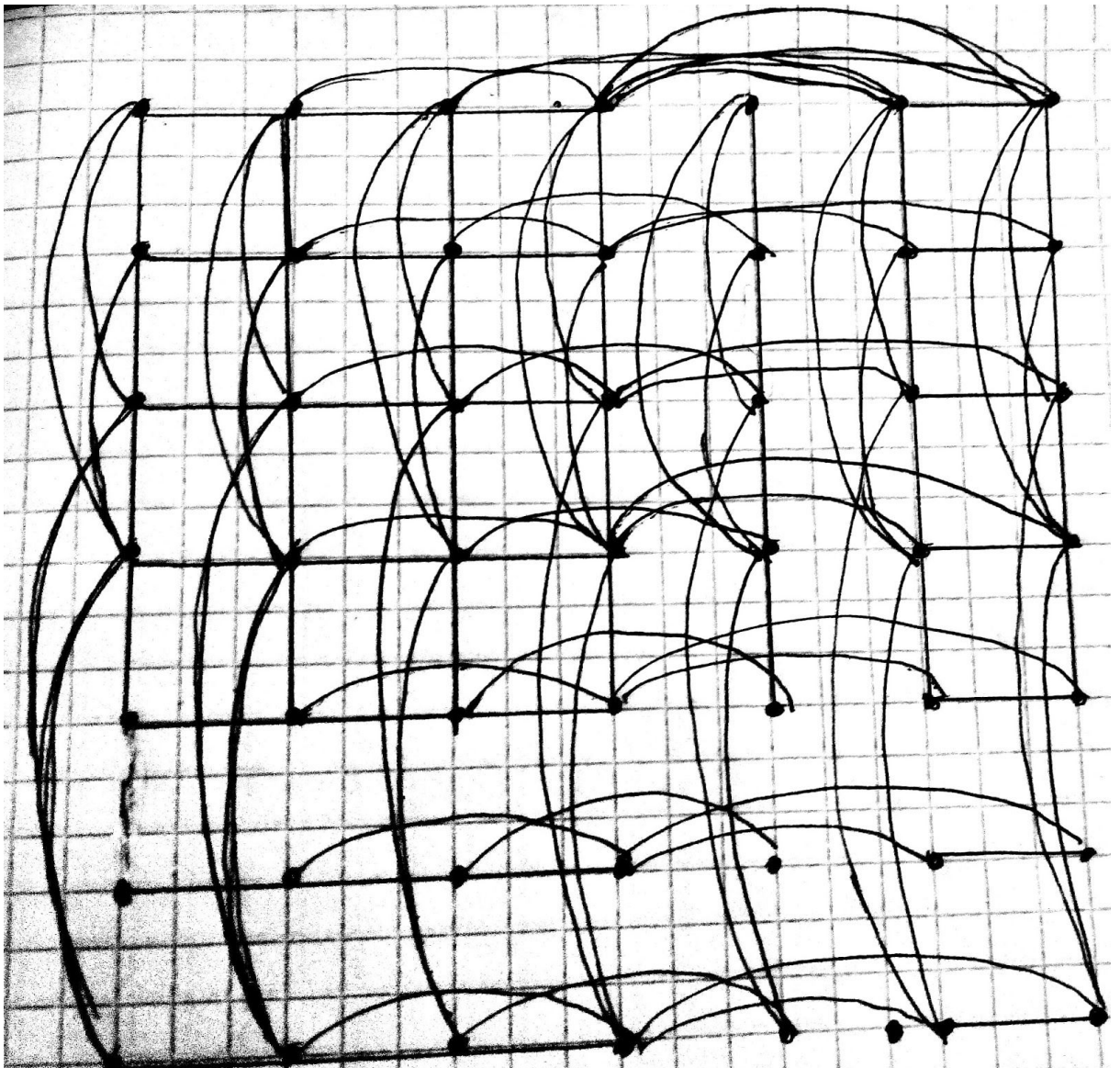
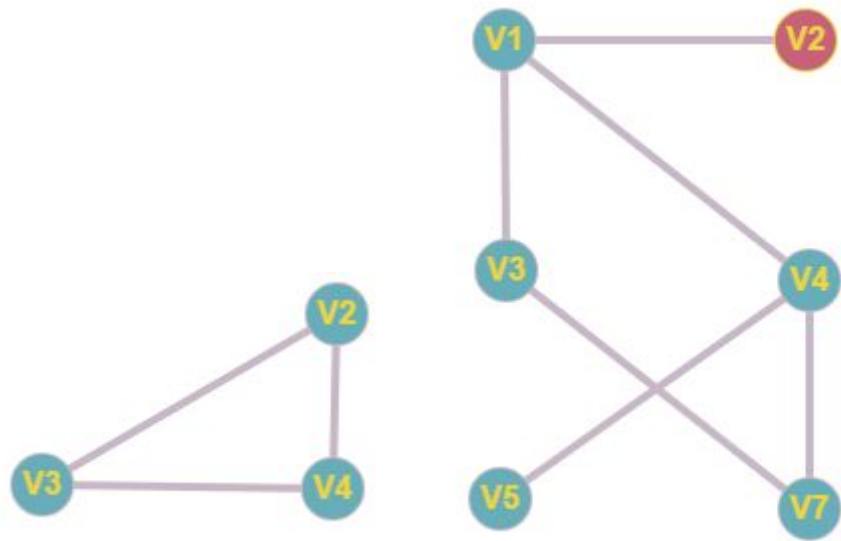
2



3

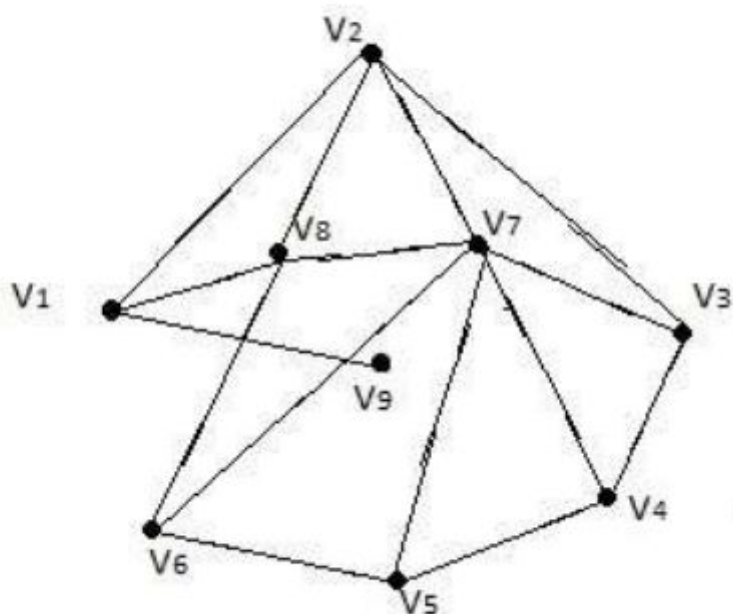


4



2. Знайти таблицю суміжності та діаметр графа

15



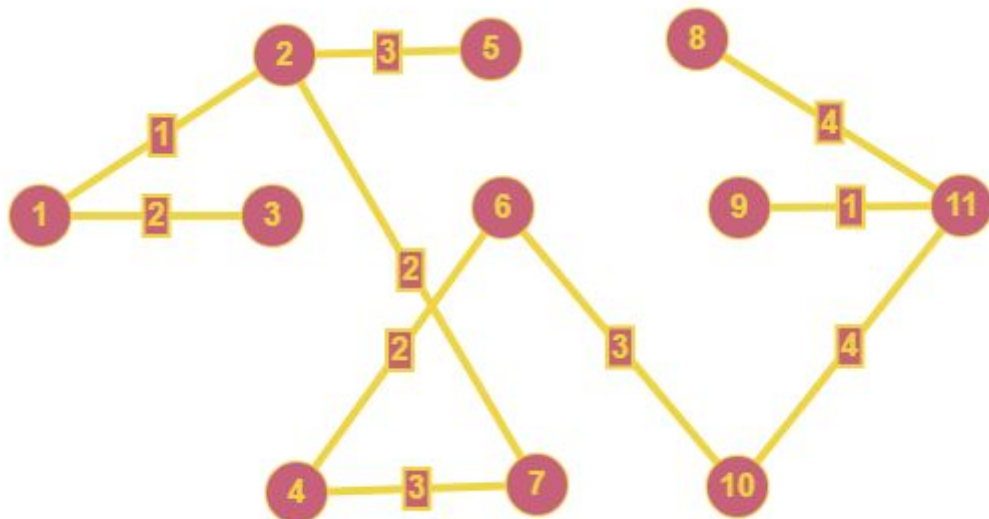
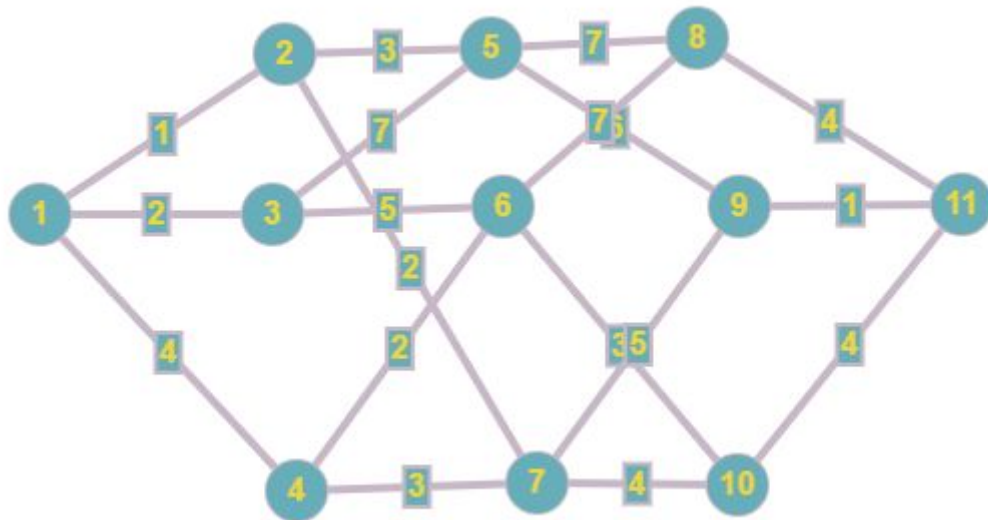
	1	2	3	4	5	6	7	8	9
1	0	1	0	0	0	0	0	1	1
2	1	0	1	0	0	1	1	1	0
3	0	1	0	1	0	0	1	0	0
4	0	0	1	0	1	0	1	0	0
5	0	0	0	1	0	1	1	0	0
6	0	1	0	0	1	0	1	1	0
7	0	1	1	1	1	1	0	1	0
8	1	1	0	0	0	1	1	0	0

9	1	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---

$d = 4$

(V9-V1-V8-V7-V3)

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



Завдання 2

```

#include<stdio.h>
#include<stdlib.h>
#define size 11

int main()
{
    int wght =0;
    int graph[size][size] = {
        { 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, 6, 0, 0, 0, 0, 0 },
        { 4, 0, 0, 0, 0, 2, 3, 0, 0, 0, 0 },
        { 0, 3, 7, 0, 0, 0, 0, 7, 5, 0, 0 },
        { 0, 0, 6, 2, 0, 0, 0, 7, 0, 3, 0 },
        { 0, 2, 0, 3, 0, 0, 0, 0, 5, 4, 0 },
        { 0, 0, 0, 0, 7, 7, 0, 0, 0, 0, 4 },
        { 0, 0, 0, 0, 5, 0, 5, 0, 0, 0, 1 },
        { 0, 0, 0, 0, 0, 3, 4, 0, 0, 0, 4 },
        { 0, 0, 0, 0, 0, 0, 0, 4, 1, 4, 0 }
    };
    int visit[size] = {0};
    int i, j, p=0, q=0;
    int arr[size]={1,2,3,4,5,6,7,8,9,10,11};
    int min;
    int flag=0;
    for(i=0;i<size;i++){
        for(j=0;j<size;j++){
            if(flag ==0 && graph[i][j]!=0){
                flag=1;
                p=i;
                q=j;
                min=graph[p][q];
            }
            else if(flag == 1 && graph[i][j]<min && graph[i][j]!=0){

```



```

        p=i;
        q=j;
        min = graph[i][j];
    }
}
}
visit[p]=1;
visit[q]=1;
int flag1=0;
int p1,q1,min1,qq=0;
printf("Ribs included i the minimum span tree \n");
printf("%d-(%d - weight of the rib)-%d ", arr[p], graph[p][q],
arr[q]);
do{
    for(i=0;i<size;i++){
        for(j=0;j<size;j++){
            if(visit[i]==1 && visit[j]==0 && graph[i][j]!=0){
                if(flag1==0){
                    flag1 = 1;
                    p1 = i;
                    q1 = j;
                    min1=graph[i][j];
                }
                else if(flag1 == 1 && graph[i][j]< min1){
                    p1 = i;
                    q1 = j;
                    min1 = graph[i][j];
                }
            }
        }
    }
}
visit[q1]=1;
flag1=0;
printf("\n%d-(%d - weight of the rib)-%d", arr[p1],
graph[p1][q1], arr[q1]);

```

```
        qq++;  
        wght = wght+graph[p1][q1];  
    }  
    while(qq < size-2);  
    printf("\n%d\n", wght);  
    return 0;  
}
```

Висновок:

я набула практичних вмінь та навичок з використання алгоритмів
Прима і Краскала.