МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

Кафедра систем штучного інтелекту

Лабораторна робота №4
з дисципліни
«Дискретна математика»

Виконав:
студент групи КН-114
Мороз Павло

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.

ТЕОРЕТИЧНІ ВІДОМОСТІ

Графом G називається пара множин (V, E), де V − множина вершин, перенумерованих числами 1, 2, ..., $n = \upsilon$; $V = \{\upsilon\}$, E − множина упорядкованих або неупорядкованих пар e = (v′,v′′), v′∈V , v′′∈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 одну і ту саму вершину, то таке ребро називається петлею.

Мультиграф – граф, який має кратні ребра.

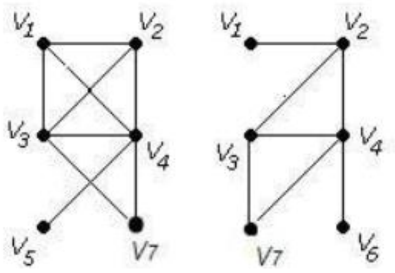
Псевдограф – граф, який має петлі.

Простий граф – граф, який не має кратних ребер та петель. Будь яке ребро e інцедентно двом вершинам (v′,v′′), які воно з’єднує. У свою чергу вершини (v′,v′′) інцендентні 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 інцедентна до жодного ребра, називається ізольованою.

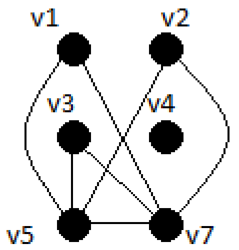
Вершина графа, яка інцедентна тільки до одного ребра, називається звисаючою. Частина $G' = (V'$, $E')$ графа $G = (V, E)$ називається підграфом графа G , якщо V′ ⊆V і E′ складається з тих і тільки тих ребер e = (v′,v′′), у яких обидві кінцеві вершини v′,v′′∈V′ . Частина $G' = (V'$, $E')$ називається суграфом або остовим підграфом графа G , якщо виконано умови: V′ =V , E′ ⊆ E .

Завдання 1

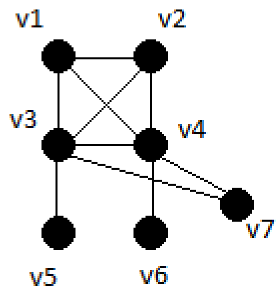
1. Виконати наступні операції над графами: 1) знайти доповнення до першого графу, 2) об’єднання графів, 3) кільцеву суму G1 та G2 (G1+G2), 4) розщепити вершину у другому графі, 5) виділити підграф A, що складається з 3-х вершин в G1 і знайти стягнення A в G1 (G1\ A), 6) добуток графів.



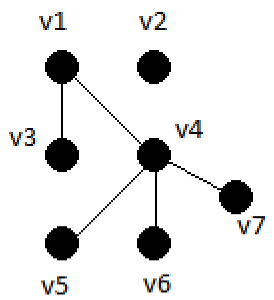
1.


v1
v2
v3
v4
v5
v7

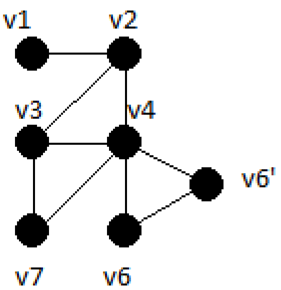
2.


v1
v2
v3
v4
v7
v5
v6

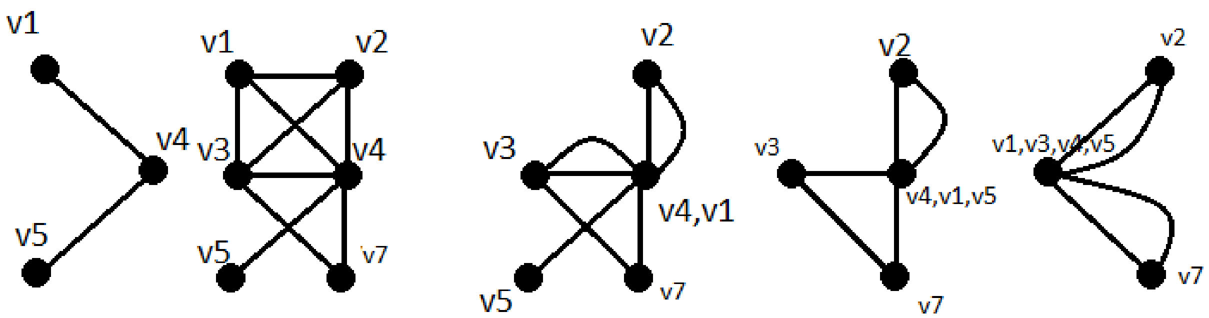
3.

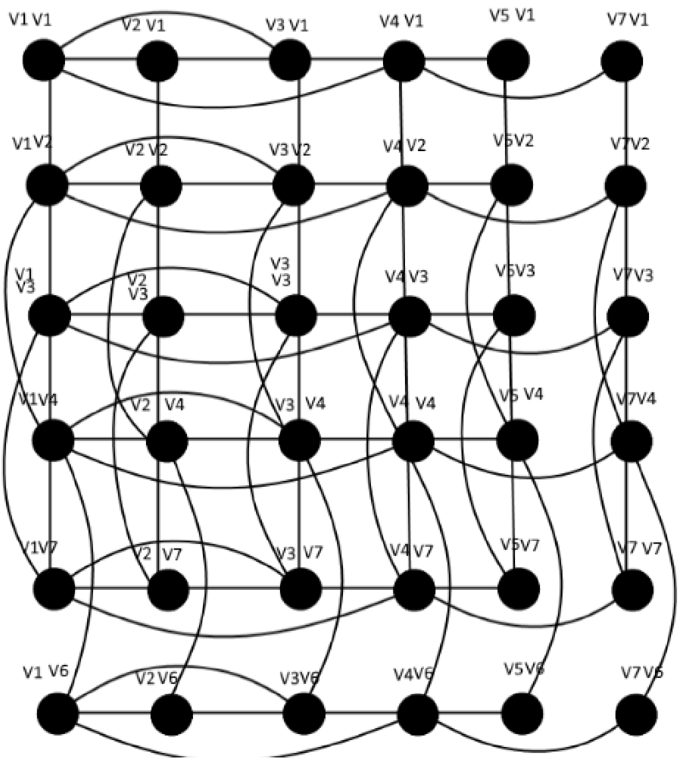

v1
v2
v3
v4
v7
v5
v6

4.


v1
v2
v3
v4
v6'
v7
v6

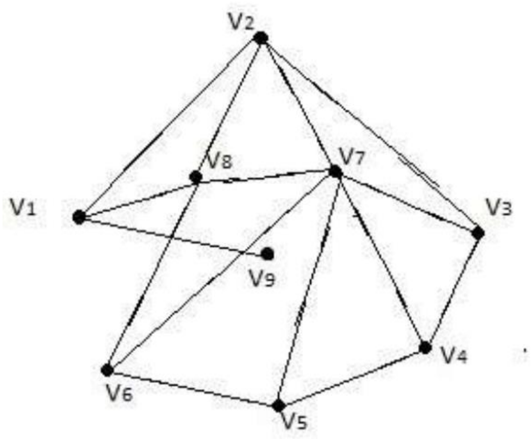
5.


v1
v4
v5
v1
v2
v3
v4
v5
v7
v2
v3
v4,v1
v5
v7
v2
v3
v4,v1,v5
v7
v2
v1,v3,v4,v5
v7


V1 V1
V2 V1
V3 V1
V4 V1
V5 V1
V7 V1
V1 V2
V2 V2
V3 V2
V4 V2
V5V2
V7V2
V1 V3
V2 V3
V3 V3
V4 V3
V5V3
V7V3
V1V4
V2 V4
V3 V4
V4 V4
V5 V4
V7V4
V1V7
V2 V7
V3 V7
V4 V7
V5V7
V7 V7
V1 V6
V2 V6
V3V6
V4V6
V5V6
V7 V6

6.

2. Знайти таблицю суміжності та діаметр графа.



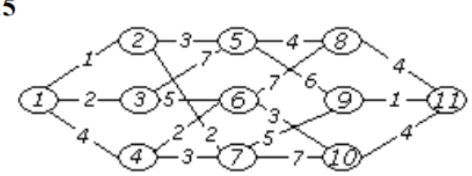
Довжина діаметра графа – 4. Він проходить по вершинах v9 – v1 – v8 – v7 – v4.

Матриця суміжності графа:

	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	1	1	0	0	0	1	1	0	0
V9	1	0	0	0	0	0	0	0	0

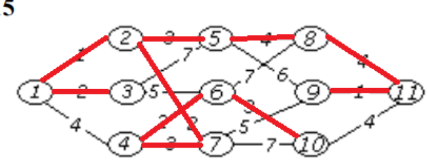
3. Знайти двома методами (Краскала і Прима) мінімальне остове дерево графа.

15



Мінімальне остове дерево графа

15

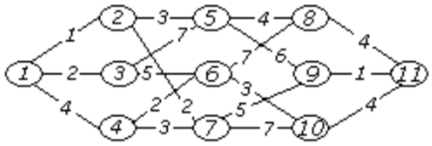


Обхід графа алгоритмом Прима з вершини 1: 1 -> 2 -> 3 -> 7 -> 4 -> 6 -> 5 -> 10 -> 8 -> 11 -> 9

Обхід графа за алгоритмом Краскалла: 1 – 2, 9 – 11, 1 – 3, 2 – 7, 4 – 6, 2 – 5, 6 – 10, 5 – 8, 8 – 11.

Завдання 2

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



Код програми

```
#include <bits/stdc++.h>

using namespace std;

 int ways[14], a[14][14],vis[14] = {0},tops[14] = {0},topNumb = 0,val,s,f;

int min_array(int n){

   int min = INT_MAX, nmin = 0;

   for(int i=1;i<=n;i++){

      if(ways[i]<min && vis[i]==0){

         min = ways[i];nmin = i;

      }

   }

   return  nmin;

}

int main() {

   int n,m;


   cout<<"Enter amount of tops and edges of graph\n";

   cin>>n>>m;

   cout<<"Enter number of start top,end top and edge weight between them\n";

   for(int i=1;i<=20;i++)

      ways[i] = INT_MAX;

   for(int i = 1;i<=m;i++){

      cin>>s>>f>>val;

      a[s][f] = val;

      a[f][s] = val;

   }
```

```
    ways[1] = 0;
    int current = 1;
    while(current!=0){
        vis[current] = 1;
        for(int i=1;i<=n;i++){
            if(a[current][i]!=0 && vis[i] ==0  )ways[i] = min(ways[i], a[current][i]);
        }
        topNumb ++;
        tops[topNumb] = current;
        current = min_array(n);
    }
    cout<<"The order of tops after algorithm passing\n";
    int sum = 0;
    for(int i=1;i<n;i++){
        sum += ways[i];
        cout<<tops[i]<<" -> ";
    }
    cout<<tops[n];
    cout<<"\nLength of minimal frame: "<<sum + ways[n];
}
```

Результат роботи програми

```
C:\Users\Admin\Desktop\Prime\bin\Debug\Prime.exe
Enter amount of tops and edges of graph
11 18
Enter number of start top,end top and edge weight between them
1 2 1
2 5 3
5 8 4
8 4 11
1 3 2
3 6 5
3 5 7
5 9 6
9 11 1
1 4 4
4 6 2
4 7 3
2 7 2
7 9 5
7 10 3
10 11 4
8 6 7
6 10 3
The order of tops after algorithm passing
1 -> 2 -> 3 -> 7 -> 4 -> 6 -> 5 -> 10 -> 8 -> 11 -> 9
Length of minimal frame: 25
Process returned 0 (0x0)   execution time : 25.152 s
Press any key to continue.
```

Висновок

Я набув практичних вмінь та навичок для роботи з графами: робити унарні та бінарні операції з графами, створювати матриці суміжності та шукати мінімальний каркас дерева алгоритмом Пріма.