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
Брила Ярослав

Викладач:
Мельникова.Н.І

Львів – 2019 р

Лабораторна робота № 4

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.

Теоритичні відомості:

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, ..., n =υ ; V = {υ}, 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 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 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 инцидентних 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 $V' \subseteq V$ 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: $V' = V$, $E' \subseteq E$.

Завдання

Варіант 2

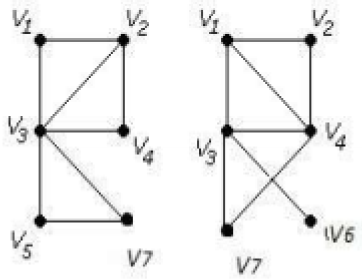
Завдання № 1. Розв'язати на графах наступні задачі:

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\ A), 6) добуток графів.

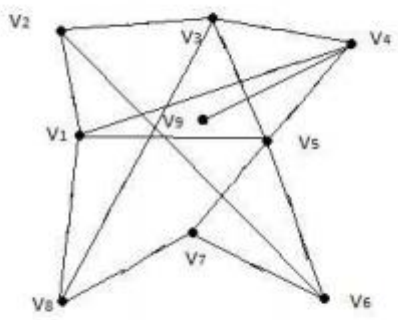
2

2



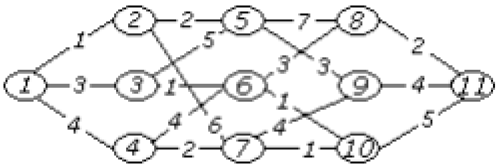
2. Знайти таблицю суміжності та діаметр графа.

2



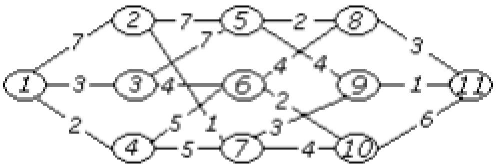
3. Знайти методами (Краскала і Прима) мінімальне остове дерево графа.

2



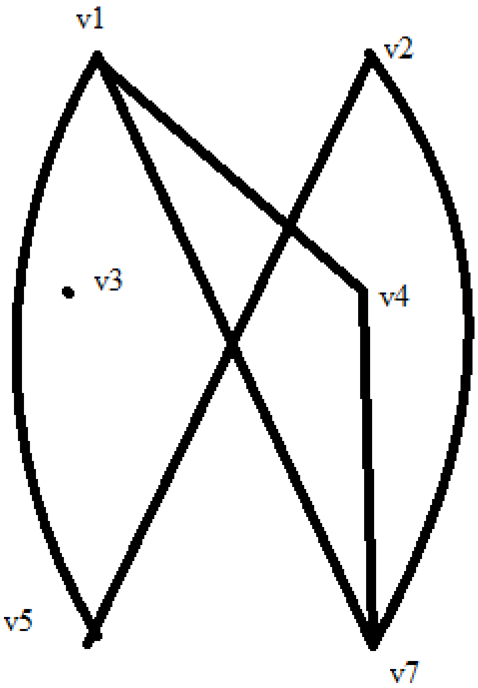
Завдання №2.

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

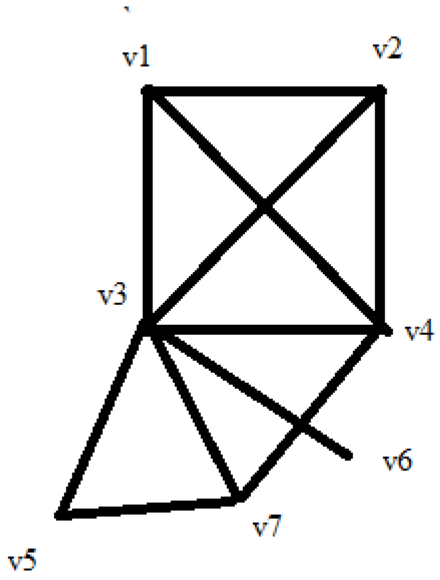


Розв’язок

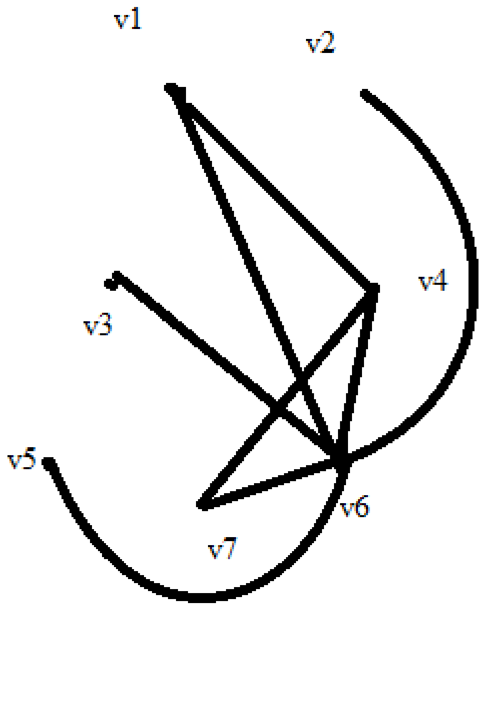
1. 1)Знайти доповнення до першого графу



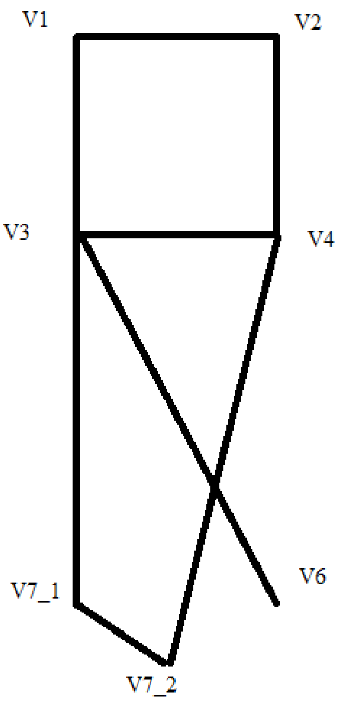
2)Знайти об’єднання графів



3)Знайти кільцеву суму G1 та G2 (G1+G2),

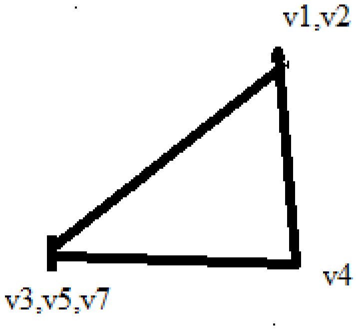


4)Розчепити вершину у другому графі

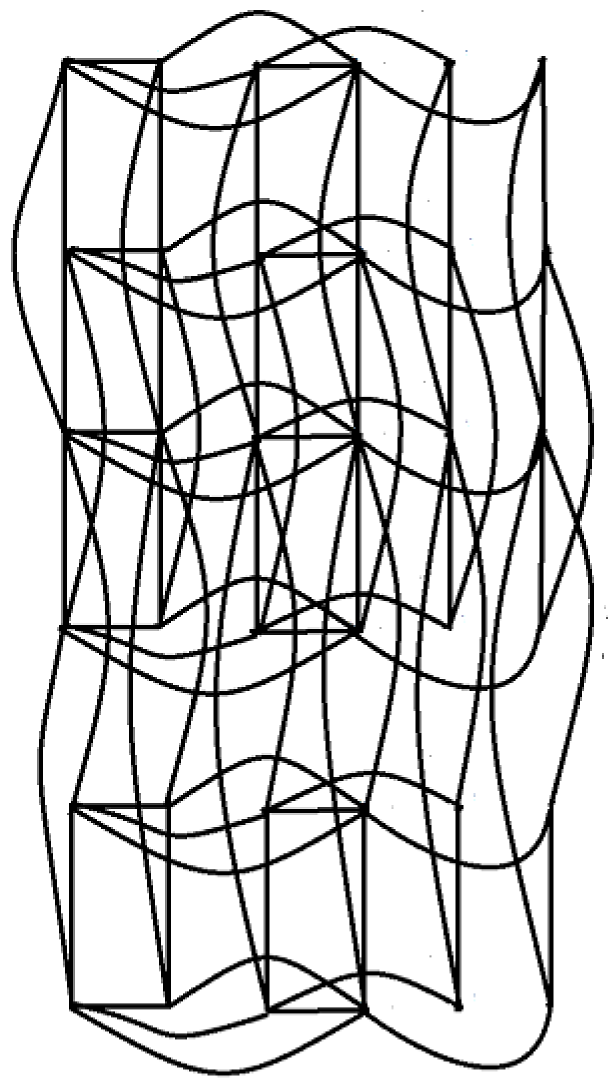


5)Виділити підграф А, що складається з 3-х вершин в G1 і знайти стягнення А в G1 (G1\ A)

Виділимо підграф(V3,V5,V7) і стягуємо (v5,v7) і (v3,v7), а потім (v3,v5),(v1,v3),(v1,v2)

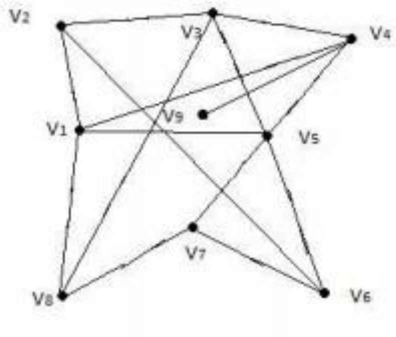


6) добуток графів



2. Знайти таблицю суміжності та діаметр графа.

2

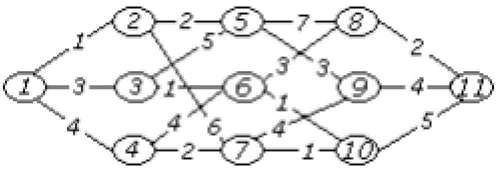


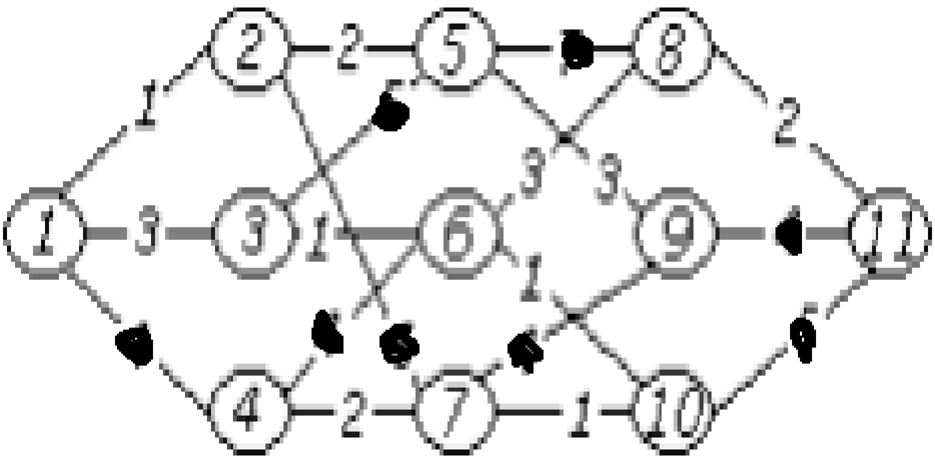
0	1	0	1	1	0	0	1	0
1	0	1	0	0	1	0	0	0
0	1	0	1	1	0	0	1	0
1	0	1	0	1	0	0	0	1
1	0	1	1	0	1	1	0	0
0	1	0	0	1	0	1	0	0
0	0	0	0	1	1	0	1	0
1	0	1	0	0	0	1	0	0
0	0	0	1	0	0	0	0	0

Діаметр графа 3(2-1-4-9)

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

2

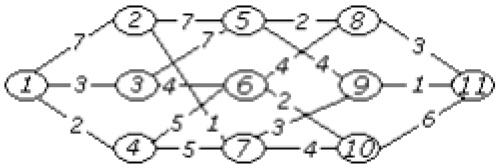




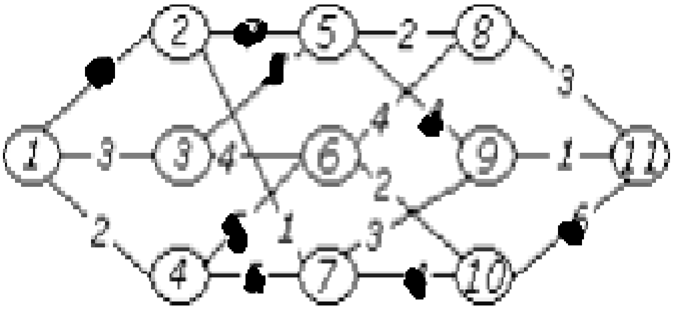
Вага=19

Завдання №2.

За алгоритмом Краскала знайти мінімальне остове дерево графа



```cpp
#include <vector>
#include <algorithm>
#include <iostream>

using namespace std;
vector<pair<int, pair<int, int> > > g; // вес - вершина 1 - вершина 2
vector<pair<int, int> > res;
int main() {
    int m,n,x,y,w,cost = 0;
    cin>>n>>m;
    vector<int> tree_id(n);

    for (int i = 0; i < m; ++i) {
        cin>>x>>y>>w; //точки та вага
        x--;y--;
        g.push_back({w,{x,y}});//вносимо в вектор
    }
    sort(g.begin(), g.end());

    for (int i = 0; i < n; ++i)
        tree_id[i] = i;

    for (int i = 0; i < m; ++i)
    {
        int a = g[i].second.first, b = g[i].second.second, l = g[i].first;
        if (tree_id[a] != tree_id[b])//порівнюємо дерева
        {
          cost += l;
            res.emplace_back(a+1, b+1);
            int old_id = tree_id[b], new_id = tree_id[a];
            for (int j = 0; j < n; ++j)
                if (tree_id[j] == old_id)
                    tree_id[j] = new_id;
        }
    }
    for(auto i:res) {
        cout<<i.first<<' '<<i.second<<endl;

    }
}
```



```
enter the number of points and the number of edges
11 18
enter points and edge weight
1 2 7
1 3 3
1 4 2
2 5 7
2 7 1
3 5 7
3 6 4
4 6 5
4 7 5
5 8 2
5 9 4
6 8 4
6 10 2
7 9 3
7 10 4
8 11 3
9 11 1
10 11 6
2 7
9 11
1 4
5 8
6 10
1 3
7 9
8 11
3 6
6 8

Weight: 25
```

Висновок

Я зміг набути практичні вміння та навички з використання алгоритмів Пріма і Краскала.