

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ
“ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра систем штучного інтелекту

Лабораторна робота №4

з дисципліни

«Дискретна математика»

Виконала:

студент групи КН-112

Хедик Адольф

Викладач:

Мельникова

Н.І.

Львів – 2019р.

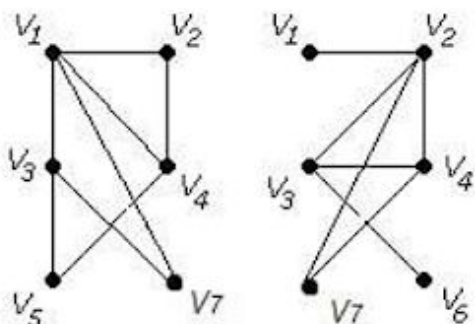
Тема: Основні операції над графами. Знаходження остова мінімальної ваги за алгоритмом Пріма-Краскала

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

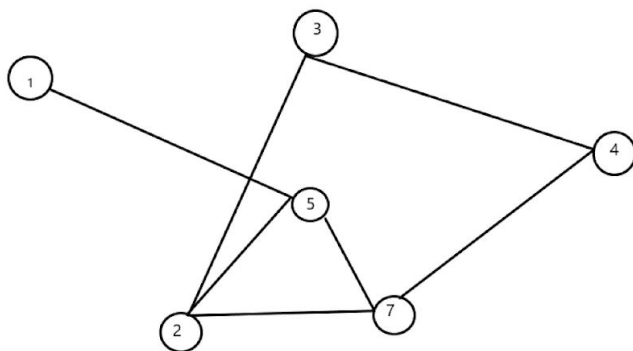
Варіант 16

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

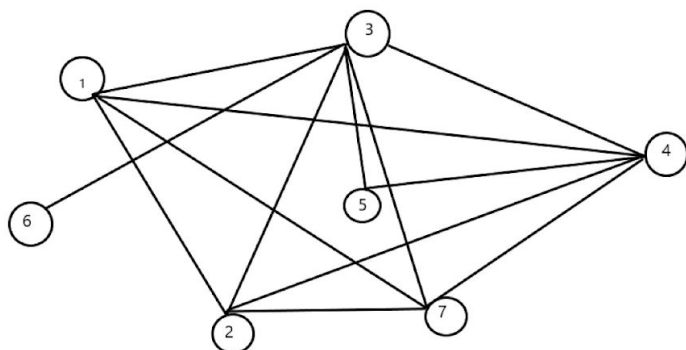
1. Виконати наступні операції над графами:



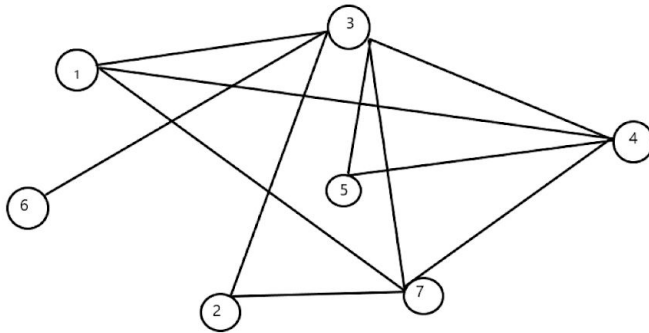
1) знайти доповнення до першого графу,



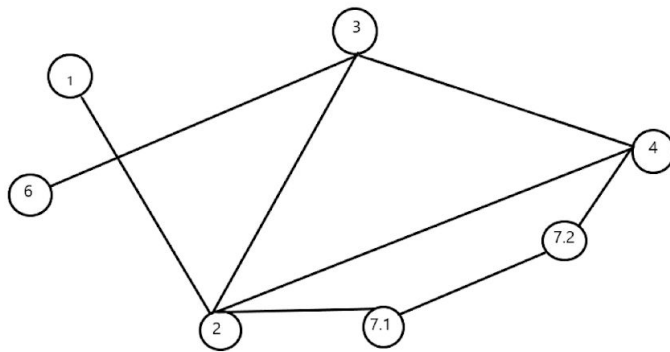
2) об'єднання графів,



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

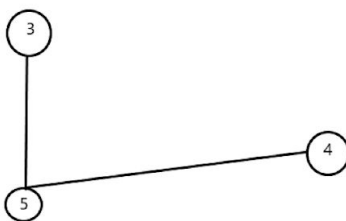


4) розщепити вершину у другому графі

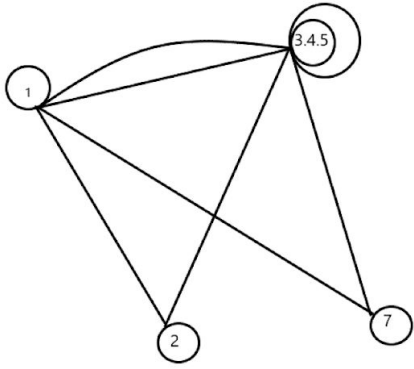


5) виділити підграф A , що складається з 3-х вершин в $G1$ і знайти стягнення A в $G1$ ($G1 \setminus A$)

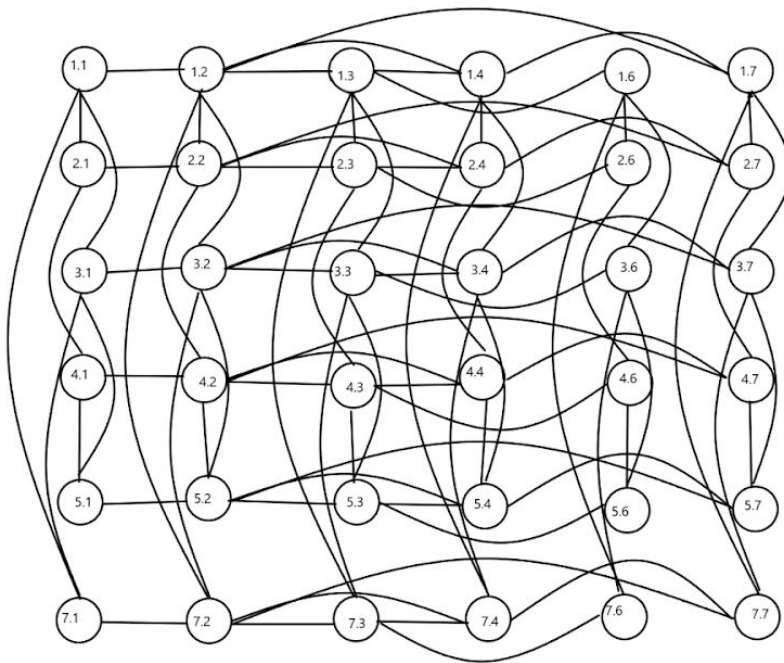
Підграф A



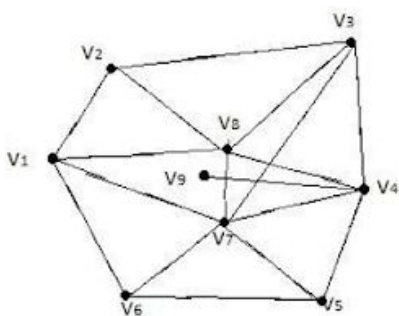
Стягнення A в $G1$



б) добуток графів.



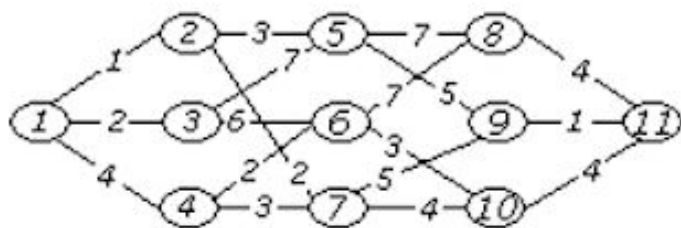
2. Знайти таблицю суміжності та діаметр графа.



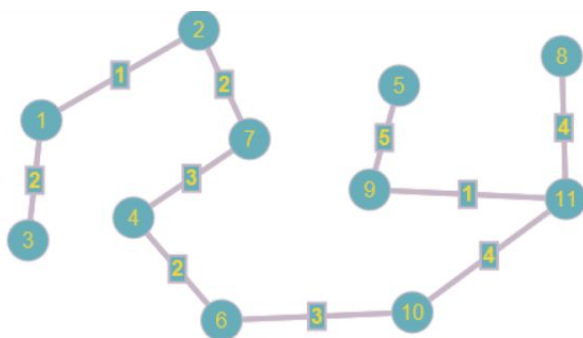
V	1	2	3	4	5	6	7	8	9
1	0	1	0	0	0	1	1	1	0
2	1	0	1	0	0	0	0	1	0
3	0	1	0	1	0	0	1	1	0
4	0	0	1	0	1	0	1	1	1
5	0	0	0	1	0	1	1	0	0
6	1	0	0	0	1	0	1	0	0
7	1	0	1	1	1	1	0	1	0
8	1	1	1	1	0	0	1	0	0
9	0	0	0	1	0	0	0	0	0

Діаметр графа дорівнює 3;

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



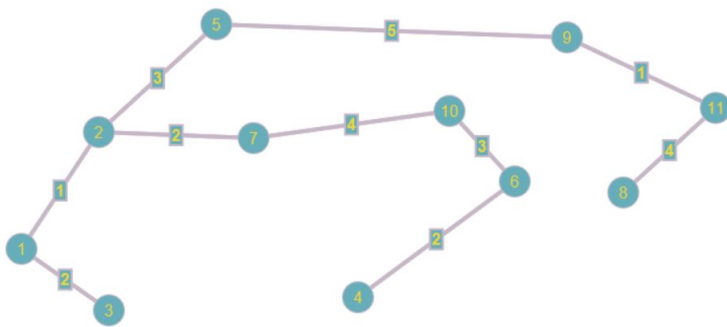
Метод Прима



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

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

Метод Краскала

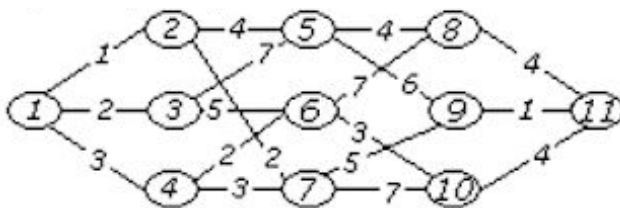


$V = \{1, 2, 9, 11, 3, 7, 4, 6, 10, 5, 8\}$

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

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

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



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

using namespace std;

struct ribb{
    int top1;
    int top2;
    int weight;
};

int main() {
    int m = 11, n = 18;
    vector <pair<int, pair<int, int>>>> g;
    ribb vals[] = {
        {1, 2, 1},
        {1, 3, 2},
        {1, 4, 3},
        {2, 5, 4},
```

```

        {2,7,2},
        {3,5,7},
        {3,6,5},
        {4,6,2},
        {4,7,3},
        {5,8,4},
        {5,9,6},
        {6,8,7},
        {6,10,3},
        {7,9,5},
        {7,10,7},
        {8,11,4},
        {9,11,1},
        {10,11,4}
    };
    for(int i=0;i<m;i++){
        vals[i].top1--;
        vals[i].top2--;
        g.push_back({vals[i].weight,{vals[i].top1,vals[i].top2}});
    }
    int cost = 0;
    vector < pair<int,int> > res;
    sort (g.begin(), g.end());
    vector<int> tree_id (n);
    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])
        {
            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 << "rib: " << i.first<< " " << i.second<< "\n";
    }
}

```

```

rib: 1 2
rib: 1 3
rib: 2 7
rib: 4 6
rib: 1 4
rib: 2 5
rib: 5 8
rib: 5 9

```

Висновок: Я набула практичних вмінь та навичок з використання алгоритмів
Пріма і Краскала.