**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

**Кафедра систем штучного інтелекту**

**Лабораторна робота №4**

З дисципліни

“Дискретна математика”

**Виконав:**

студент групи КН-112

Думич Іван

**Викладач:**

Мельникова Н.І.

Львів – 2019 р.

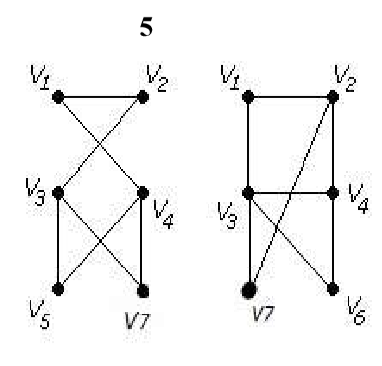
**Тема:** Основні операції над графами. Знаходження остова мінімальної ваги за алгоритмом Прима-Краскала.

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

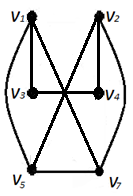
**Варіант № 5**

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

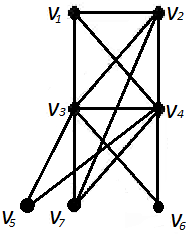
1. Виконати наступні операції над графами:



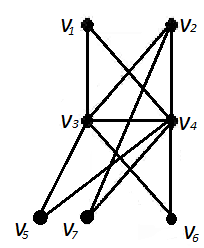
1) Знайти доповнення до першого графу.



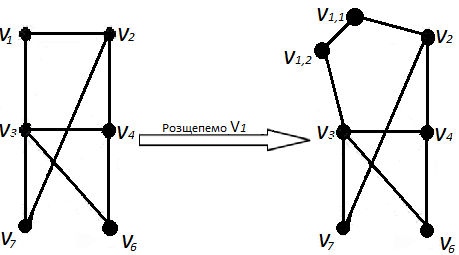
2) Об’єднання графів.



3) Кільцеву суму G1 та G2 (G1+G2).

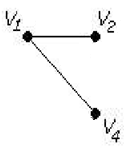


4) Розщепити вершину у другому графі.

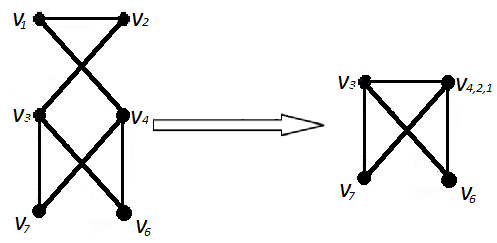


5) Виділити підграф А, що складається з 3-х вершин в G1 і знайти стягнення А в G1 (G1\ A).

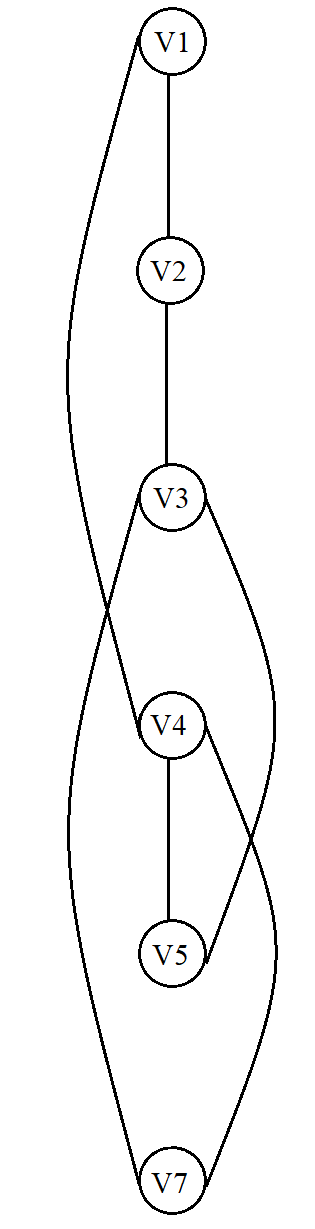
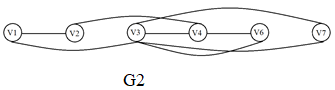
Підграф A:

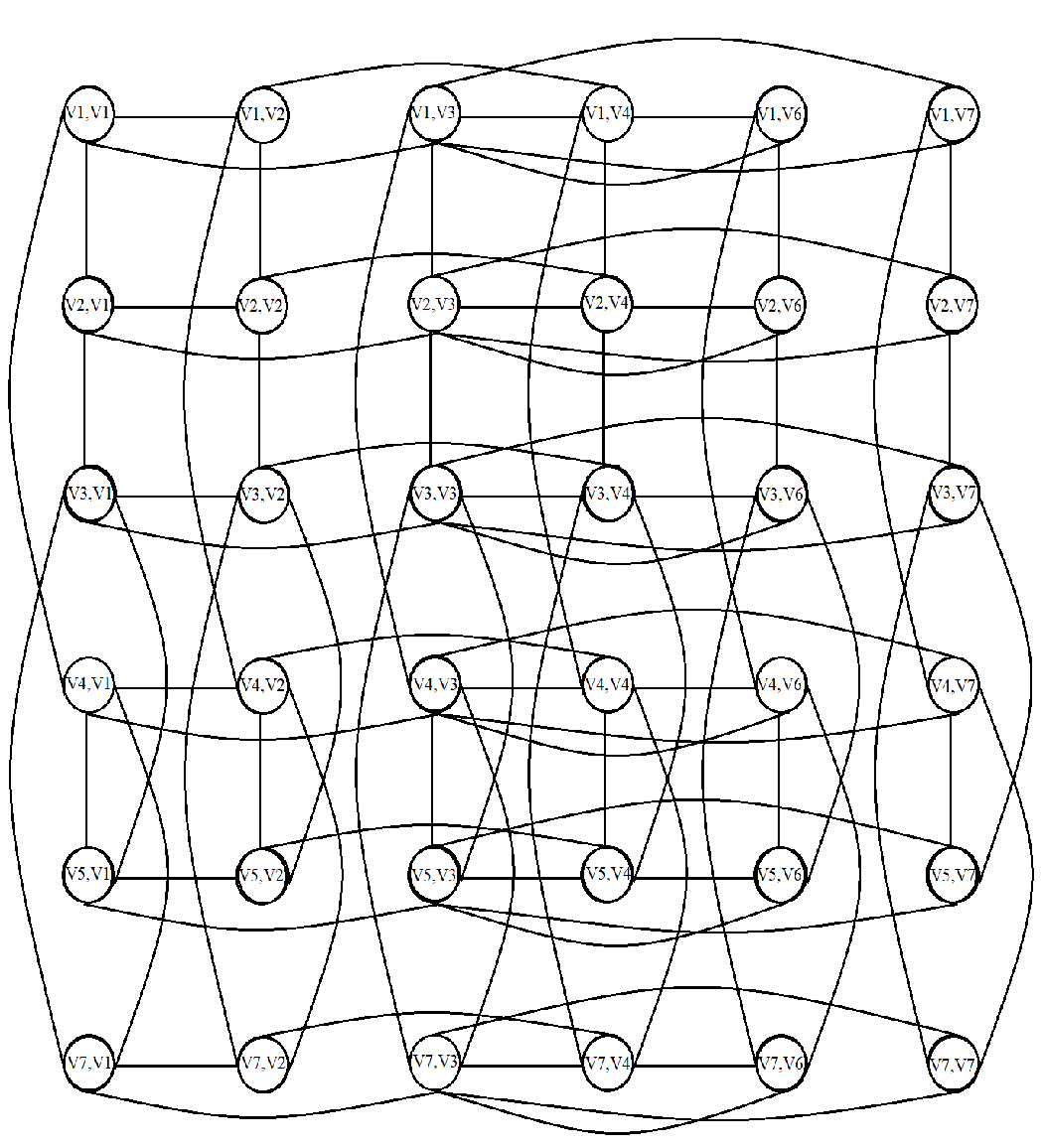


Cтягнення А в G1:

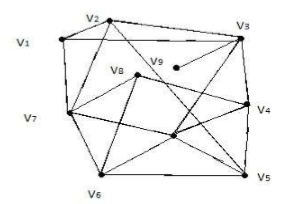


6) Добуток графів.

G1



2. Знайти таблицю суміжності та діаметр графа.



Таблиця суміжності

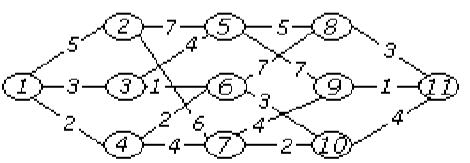
|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 | V9 | V10 |
| V1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| V2 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| V3 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| V4 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| V5 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| V6 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| V7 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| V8 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| V9 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| V10 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |

Побудуємо матрицю відстаней між вершинами

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 | V9 | V10 |
| V1 | 0 | 1 | 1 | 2 | 2 | 2 | 1 | 2 | 2 | 2 |
| V2 |  | 0 | 1 | 2 | 1 | 2 | 1 | 2 | 2 | 2 |
| V3 |  |  | 0 | 1 | 2 | 2 | 2 | 2 | 1 | 1 |
| V4 |  |  |  | 0 | 1 | 2 | 2 | 1 | 2 | 1 |
| V5 |  |  |  |  | 0 | 1 | 2 | 2 | 3 | 1 |
| V6 |  |  |  |  |  | 0 | 1 | 1 | 0 | 1 |
| V7 |  |  |  |  |  |  | 0 | 1 | 3 | 1 |
| V8 |  |  |  |  |  |  |  | 0 | 3 | 2 |
| V9 |  |  |  |  |  |  |  |  | 0 | 2 |
| V10 |  |  |  |  |  |  |  |  |  | 0 |

Максимальне число 3. Отже Діаметр графа дорівнює 3.

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



Методом Прима:

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

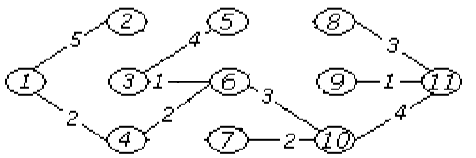
E = {(1,4), (4,6), (6,3), (6,10), (10,7), (10,11), (11,9), (11,8), (3,5), (1,2)}

Методом Краскала:

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

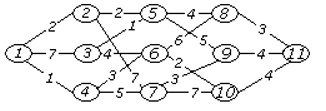
E = {3,6), (9,11), (7,10), (1,4), (4,6), (6,10), (11,8), (10,11), (3,5), (1,2)}

Мінімальне остове дерево має вигляд:



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

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



**Програмна реалізація:**

#include <stdio.h>

#include <stdlib.h>

#include <iostream>

using namespace std;

int main()

{

    setlocale(LC\_ALL, "Ukrainian");

    int v, count = 0, min = 0, k, t;

    bool check =false;

    cout << "Кiлькiсть вершин графа : ";

    cin >> v;

    int\* tops = new int[v];

    int\*\* graph = new int\* [v];

    int\*\* ribs = new int\* [v - 1];

    for (int j = 0; j < v; j++) {

        graph[j] = new int[v];

    }

    for (int j = 0; j < v - 1; j++) {

        ribs[j] = new int[2];

    }

    for (int a = 0; a < v; a++) {

        for (int j = 0; j < v; j++) {

            cin >> graph[a][j];

        }

    }

    //////Будуємо дерево, що включає в себе одну вершину

    tops[count] = 1;

    count++;

     ///Знаходження мінімального кістякового дерева

    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 && graph[tops[j] - 1][a] > 0) {

                    min = graph[tops[j] - 1][a];

                    k = ribs[count - 1][0] = tops[j]; t = ribs[count - 1][1] = a + 1;

                    continue;

                }

                if (graph[tops[j] - 1][a] > 0 && graph[tops[j] - 1][a] < min) {

                    min = graph[tops[j] - 1][a];

                    k=ribs[count-1][0] = tops[j]; t = ribs[count-1][1] = a + 1;

                }

            }

        }

        graph[k-1][t-1] = 0; graph[t-1][k-1] = 0;

        tops[count] = t;

        count++;

        min = 0;

    }

    /////Результат

    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 << "( " << ribs[j][0]<<", "<< ribs[j][1]<<" ), ";

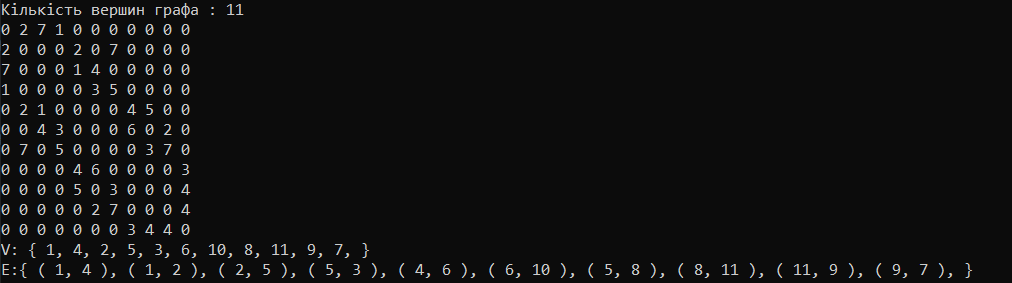
    }

    cout << "}";

        return 0;

    }

**Результат:**



**Висновок**: я набув практичних вмінь та навичок з використання алгоритмів Прима і Краскала.