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

**Кафедра систем штучного інтелекту**

**Лабораторна робота №4**

з дисципліни

«Дискретна математика»

**Виконав:**

студент групи КН-113

Саланчій Тарас

**Викладач:**

Мельникова Н.І.

Львів – 2019р.

**Тема:** Основні операції над графами. Знаходження остова

мінімальної ваги за алгоритмом Пріма-Краскала

**Мета:** набуття практичних вмінь та навичок з використання

алгоритмів Пріма і Краскала.

**Варіант №10**

**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 A та B (A+B),

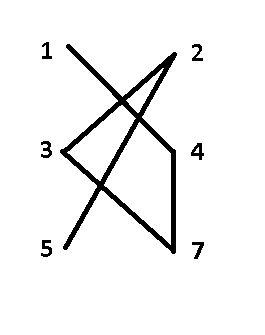
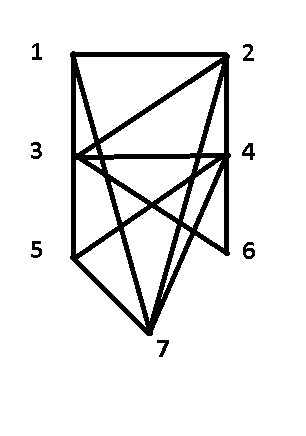
4) розщепити вершину у другому графі,

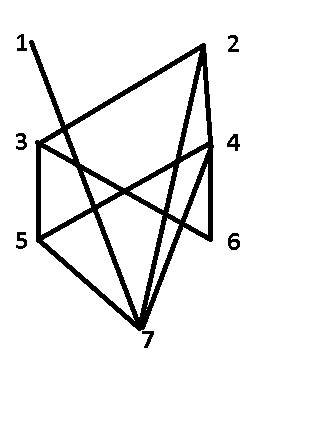
5) виділити підграф G, що складається з 3-х вершин в A і знайти

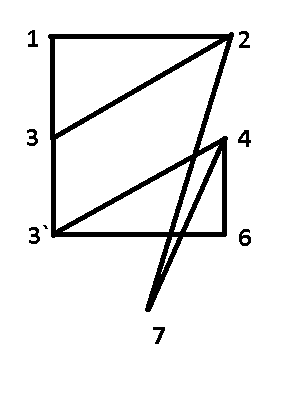
стягнення G в A (A\ G),

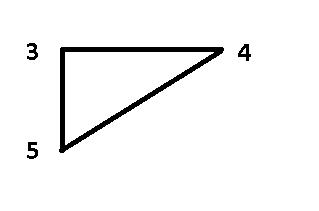
6) добуток графів.

A B

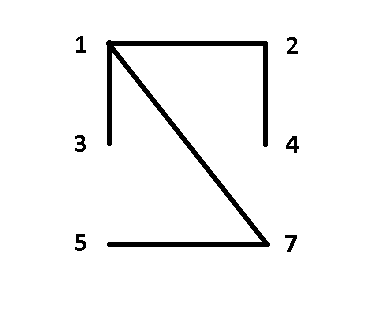




Розщеплення вершини 3 графу 2

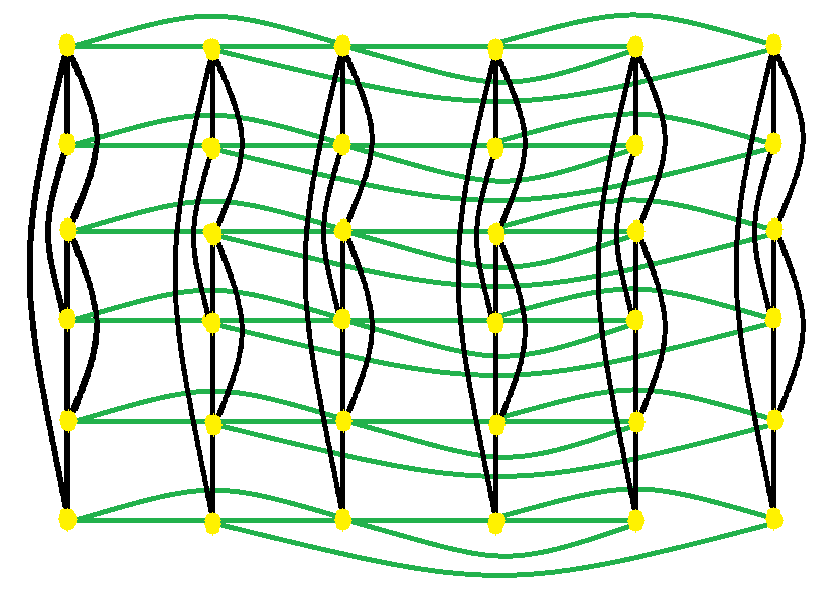
Граф G

Aaa

A\G

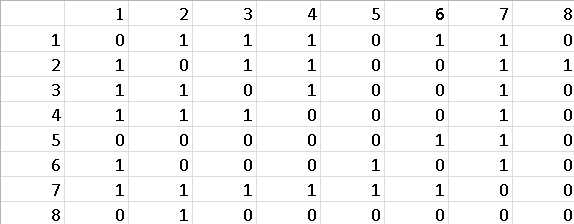
Добуток графів A\*B





**2.** Знайти таблицю суміжності та діаметр графа.

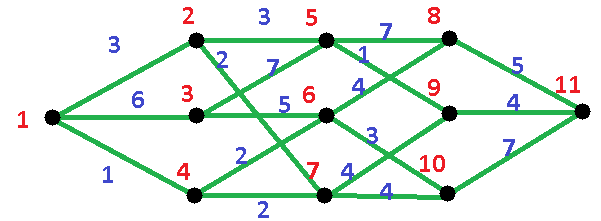
Таблиця суміжності



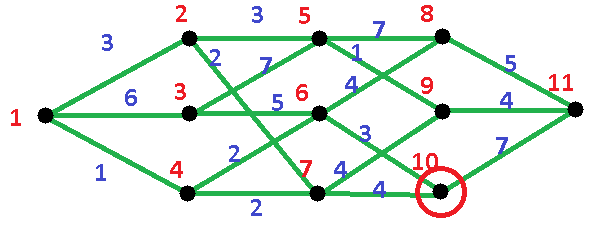
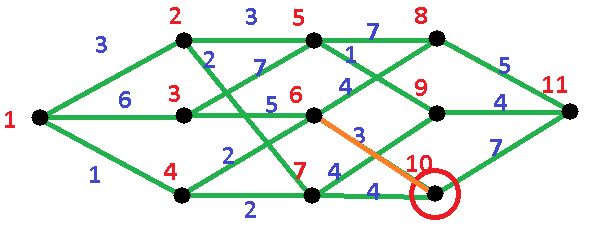
Діаметр графа – 3.

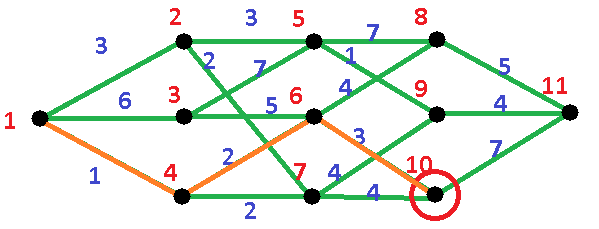
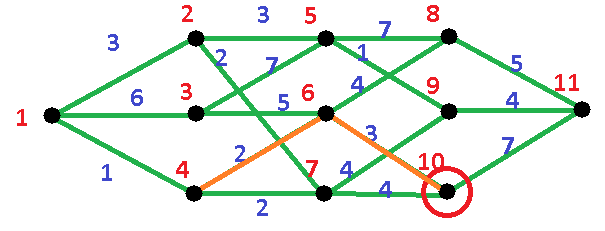
**3.** Знайти двома методами (Краскала і Прима) мінімальне остове

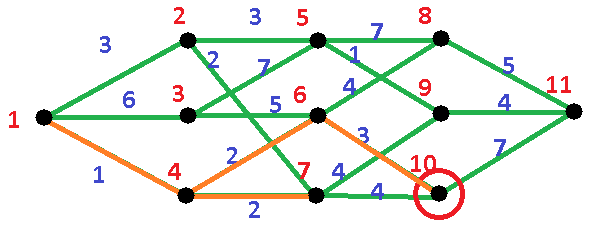
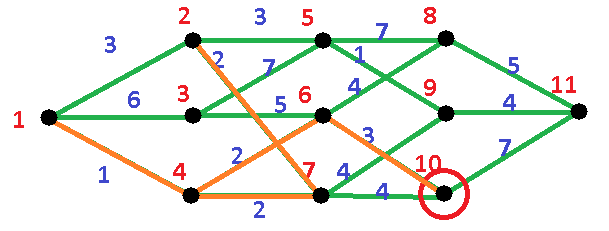
дерево графа.

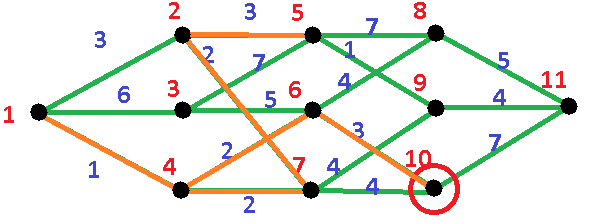
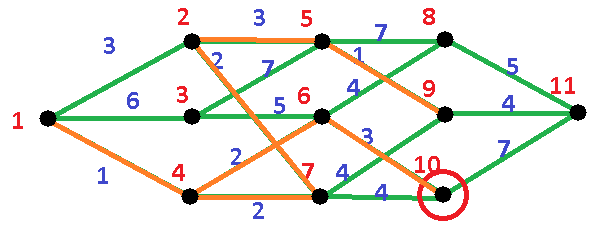


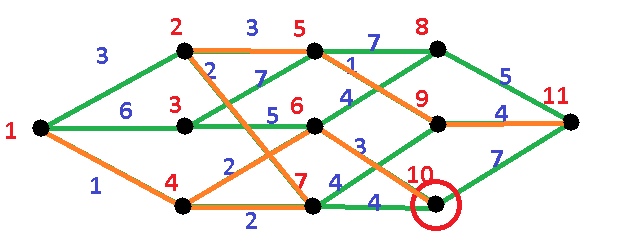
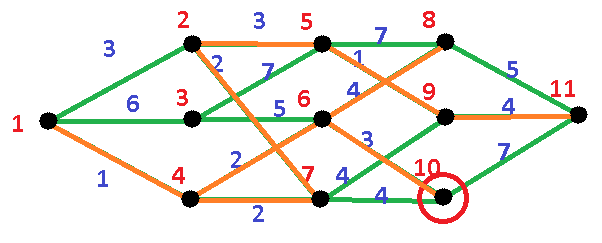
Прима

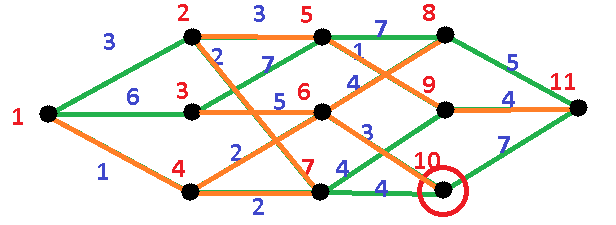


****

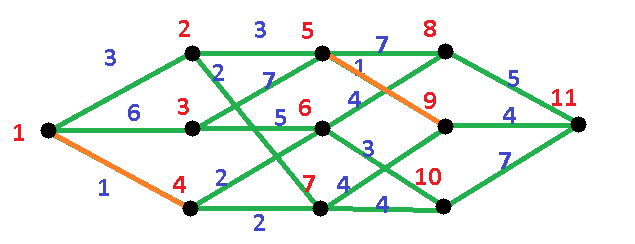
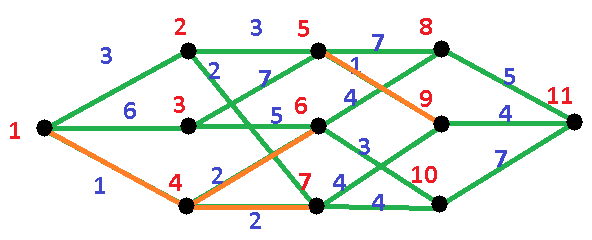
****

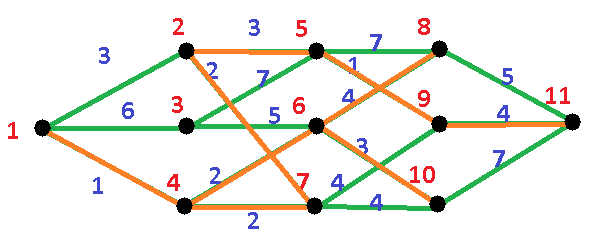
****

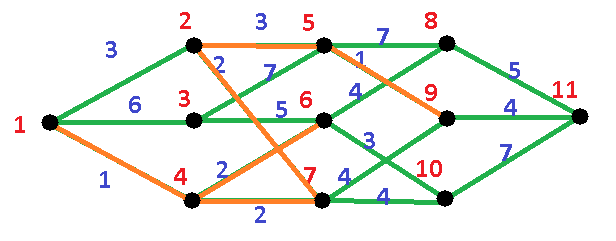
****

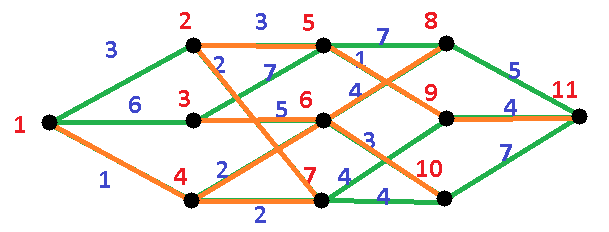
****

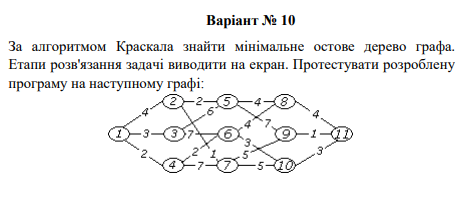
**Краскала**

****

****

****

****



**Код програми**

**#include <iostream>**

**#include <stdio.h>**

**using namespace std;**

**struct mas**

**{**

**int crus[11] = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };**

**int c = 0;**

**};**

**struct rebro**

**{**

**int len;**

**int a;**

**int b;**

**bool in = 0;**

**};**

**int**

**in (rebro \* reb, int n)**

**{**

**for (int i = 0; i < n; i++)**

**{**

**cout << "first edge of" << i + 1 << " reb: ";**

**cin >> reb[i].a;**

**cout << "second edge of " << i + 1 << " reb: ";**

**cin >> reb[i].b;**

**cout << "weight of " << i + 1 << " reb: ";**

**cin >> reb[i].len;**

**cout << endl;**

**}**

**return 0;**

**}**

**int**

**main ()**

**{**

**int n , x = 7, y = 7 ,z;**

**n=18,z=11;**

**rebro \*reb = new rebro[n];**

**mas inn[5];**

**reb[0].a=1;reb[0].b=2;reb[0].len=4;**

**//---------------------------------**

**reb[1].a=1;reb[1].b=3;reb[1].len=3;**

**//---------------------------------**

**reb[2].a=1;reb[2].b=4;reb[2].len=2;**

**//---------------------------------**

**reb[3].a=2;reb[3].b=5;reb[3].len=2;**

**//---------------------------------**

**reb[4].a=2;reb[4].b=7;reb[4].len=1;**

**//---------------------------------**

**reb[5].a=3;reb[5].b=5;reb[5].len=6;**

**//---------------------------------**

**reb[6].a=3;reb[6].b=6;reb[6].len=7;**

**//---------------------------------**

**reb[7].a=4;reb[7].b=6;reb[7].len=2;**

**//---------------------------------**

**reb[8].a=4;reb[8].b=7;reb[8].len=7;**

**//---------------------------------**

**reb[9].a=5;reb[9].b=8;reb[9].len=4;**

**//---------------------------------**

**reb[10].a=5;reb[10].b=9;reb[10].len=7;**

**//---------------------------------**

**reb[11].a=6;reb[11].b=8;reb[11].len=4;**

**//---------------------------------**

**reb[12].a=6;reb[12].b=10;reb[12].len=3;**

**//---------------------------------**

**reb[13].a=7;reb[13].b=9;reb[13].len=5;**

**//---------------------------------**

**reb[14].a=7;reb[14].b=10;reb[14].len=5;**

**//---------------------------------**

**reb[15].a=8;reb[15].b=11;reb[15].len=4;**

**//---------------------------------**

**reb[16].a=9;reb[16].b=11;reb[16].len=1;**

**//---------------------------------**

**reb[17].a=10;reb[17].b=11;reb[17].len=3;**

**// in (reb, n);**

**for (int i = 0; i < n - 1; i++)**

**{**

**for (int j = 0; j < n - 1; j++)**

**{**

**if (reb[j].len > reb[j + 1].len)**

**{**

**swap (reb[j].len, reb[j + 1].len);**

**swap (reb[j].a, reb[j + 1].a);**

**swap (reb[j].b, reb[j + 1].b);**

**}**

**}**

**}**

**int c = -1;**

**for (int i = 0; i < n; i++)**

**{**

**for (int j = 0; j < 5; j++)**

**{**

**for (int k = 0; k < z; k++)**

**{**

**if (reb[i].a == inn[j].crus[k])**

**{**

**x = j;**

**goto point0;**

**}**

**}**

**}**

**point0:;**

**for (int j = 0; j < 5; j++)**

**{**

**for (int k = 0; k < z; k++)**

**{**

**if (reb[i].b == inn[j].crus[k])**

**{**

**y = j;**

**goto point1;**

**}**

**}**

**}**

**point1:;**

**if (x != y && x == 7)**

**{**

**inn[y].crus[inn[y].c] = reb[i].a;**

**inn[y].c++;**

**}**

**if (x != y && y == 7)**

**{**

**inn[x].crus[inn[x].c] = reb[i].b;**

**inn[x].c++;**

**}**

**if (x != y && x != 7 && y != 7)**

**{**

**if (x < y)**

**{**

**for (int l = 0; l < inn[y].c; l++)**

**{**

**inn[x].crus[inn[x].c + l] = inn[y].crus[l];**

**inn[y].crus[l] = 0;**

**}**

**inn[x].c += inn[y].c;**

**inn[y].c = 0;**

**}**

**if (y < x)**

**{**

**for (int l = 0; l < inn[x].c; l++)**

**{**

**inn[y].crus[inn[y].c + l] = inn[x].crus[l];**

**inn[x].crus[l] = 0;**

**}**

**inn[y].c += inn[x].c;**

**inn[x].c = 0;**

**}**

**}**

**if (x == 7 && y == 7)**

**{**

**c++;**

**inn[c].crus[inn[c].c] = reb[i].a;**

**inn[c].crus[inn[c].c + 1] = reb[i].b;**

**inn[c].c += 2;**

**}**

**reb[i].in = 1;**

**if (x == y && x != 7)**

**{**

**reb[i].in = 0;**

**}**

**x = 7;**

**y = 7;**

**}**

**cout << "steps: " << endl;**

**int s = 0;**

**for (int i = 0; i < n; i++)**

**{**

**if (reb[i].in == 1)**

**{**

**cout << "reb" << " " << reb[i].a << " -- " << reb[i].b<< " " << "weight=" <<reb[i].len << endl;**

**}**

**}**

**return 0;**

**}**

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