МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Інститут комп'ютерних наук та інформаційних технологій

Кафедра систем штучного інтелекту

Лабораторна робота №4

З дисципліни

«Дискретна математика»

Виконав:

студент групи КН-113

Рябчук Андрій

Викладач:

Мельникова Н.І.

Львів-2019

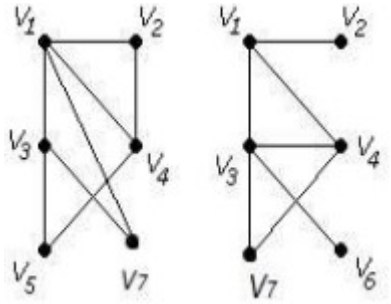
Тема: Основні операції над графами. Знаходження остова мінімальної ваги за алгоритмом Пріма-Краскала

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

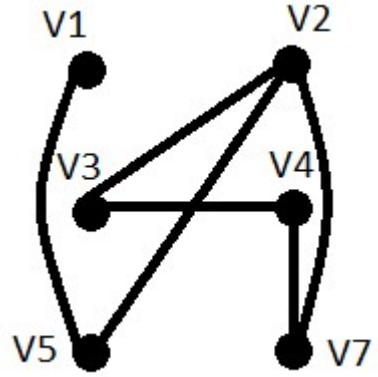
Варіант №9

Завдання № 1.

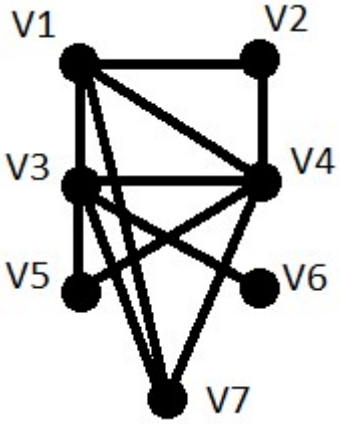
Розв'язати на графах наступні задачі: 1. Виконати наступні операції над графами: 1) знайти доповнення до першого графу, 2) об’єднання графів, 3) кільцеву суму G1 та G2 (G1+G2), 4) розщепити вершину у другому графі, 5) виділити підграф А, що складається з 3-х вершин в G1 і знайти стягнення А в G1 (G1\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



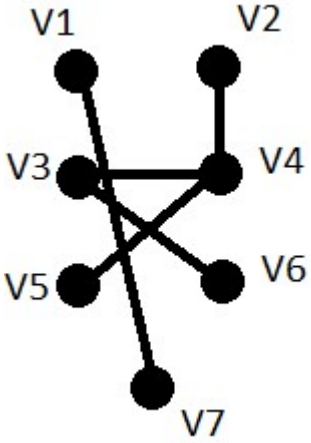
1)



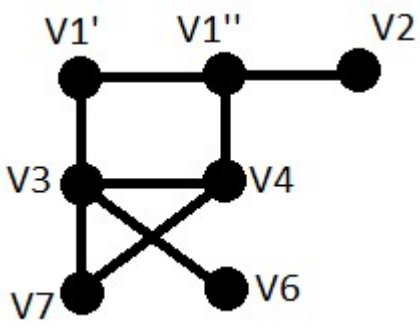
2)



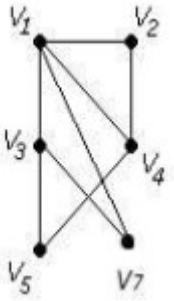
3)

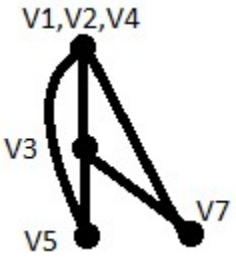


4)

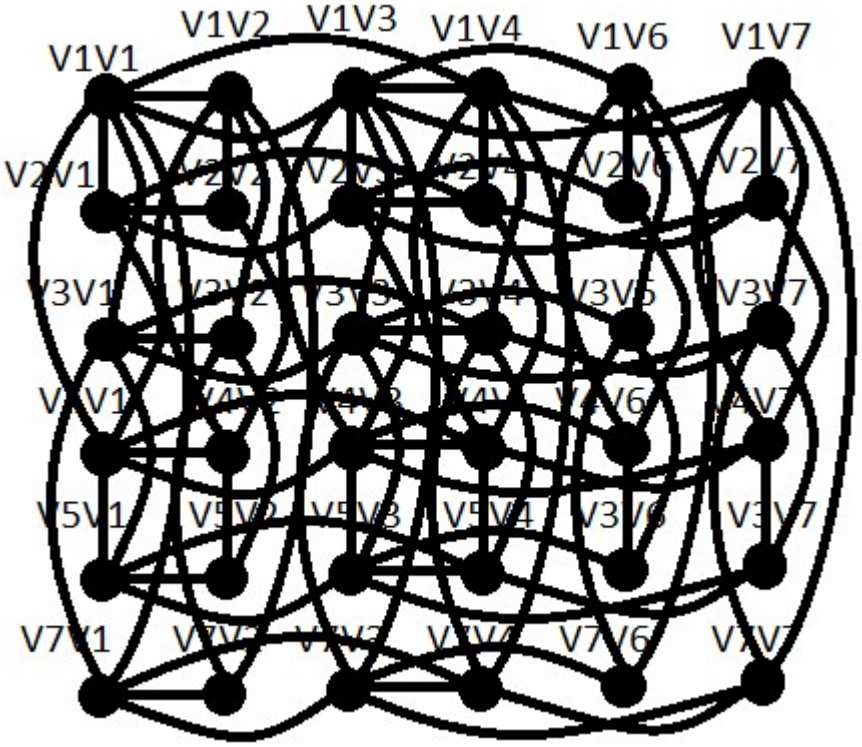


5)

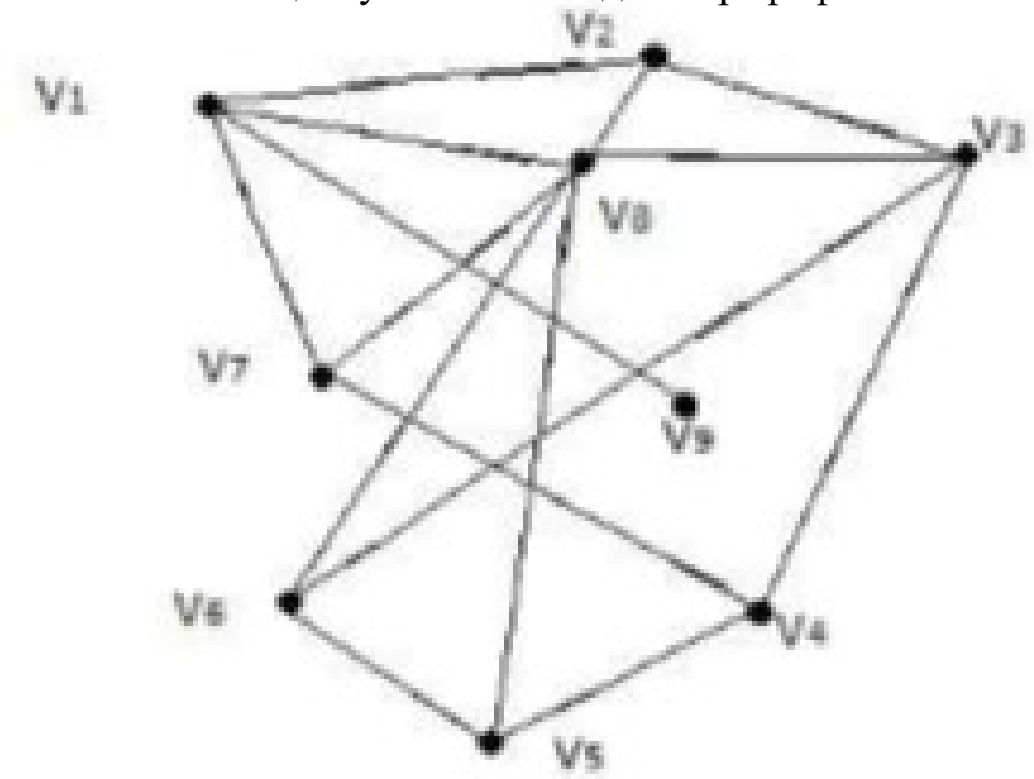

V1
V2
V3
V4
V5
V7


V1,V2,V4
V3
V7
V5

6)


V1V1
V1V2
V1V3
V1V4
V1V6
V1V7
V2V1
V3V1
V3V7
V5V1
V7V1

2. Знайти таблицю суміжності та діаметр графа



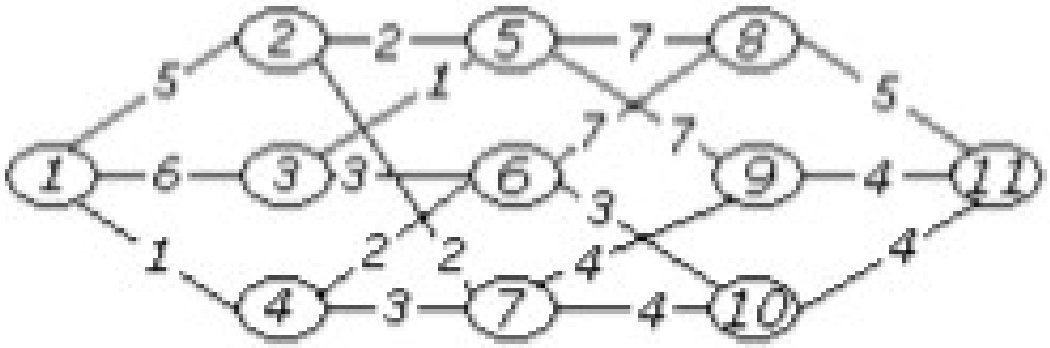
	V1	V2	V3	V4	V5	V6	V7	V8	V9
V1	0	1	0	0	0	0	1	1	1
V2	1	0	1	0	0	0	0	1	0
V3	0	1	0	1	0	1	0	1	0
V4	0	0	1	0	1	0	1	0	0
V5	0	0	0	1	0	1	0	1	0
V6	0	0	1	0	1	0	0	1	0
V7	1	0	0	1	0	0	0	1	0
V8	1	1	1	0	1	1	1	0	0
V9	1	0	0	0	0	0	0	0	0

D = 3

\

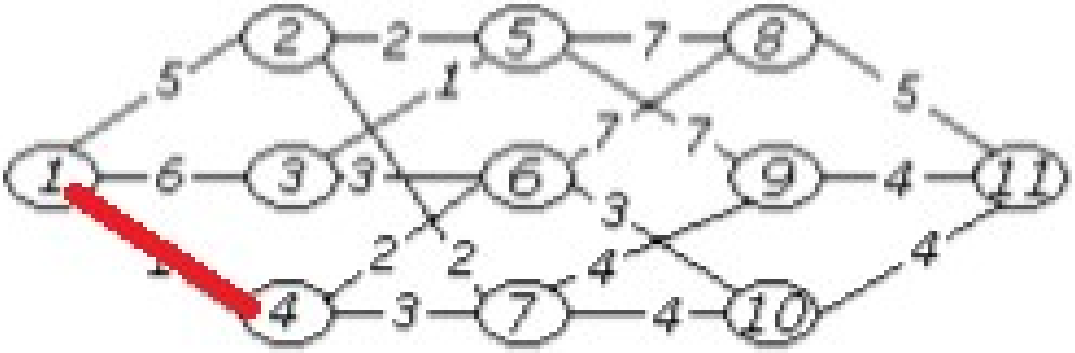
3. Знайти двома методами (Краскала і Прима) мінімальне

остове дерево графа.

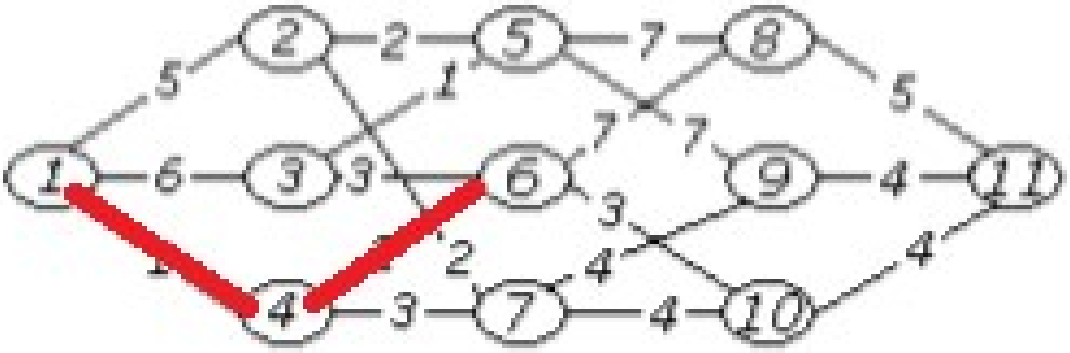


Алгоритм Прима

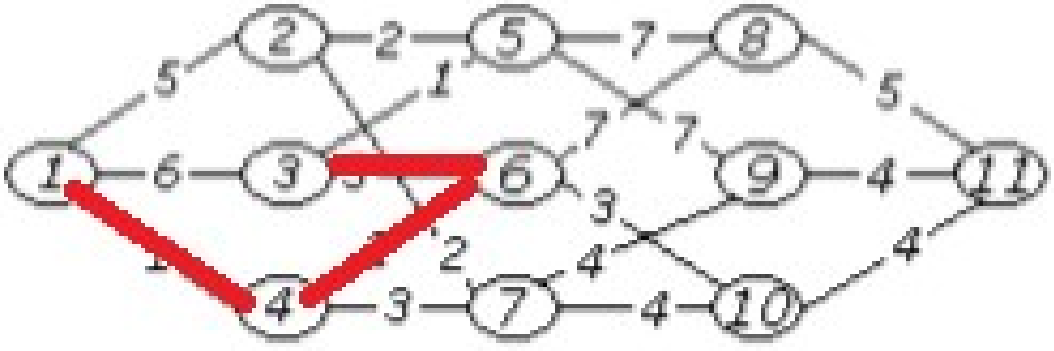
1. Як початкову вибиремо 1, далі вибираємо наближчу вершину до вершиини 1,додаємо вершину 4 до нашого дерева



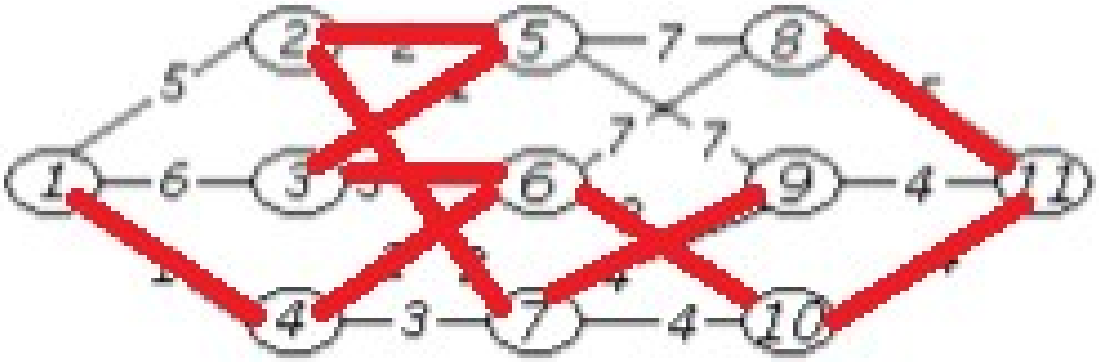
2. Далі вибираємо найближчу вершину до будь якої з двох вже присутніх.



3. В нас є можливість вибрати три вершини 3,10,7.
Вибиремо 3

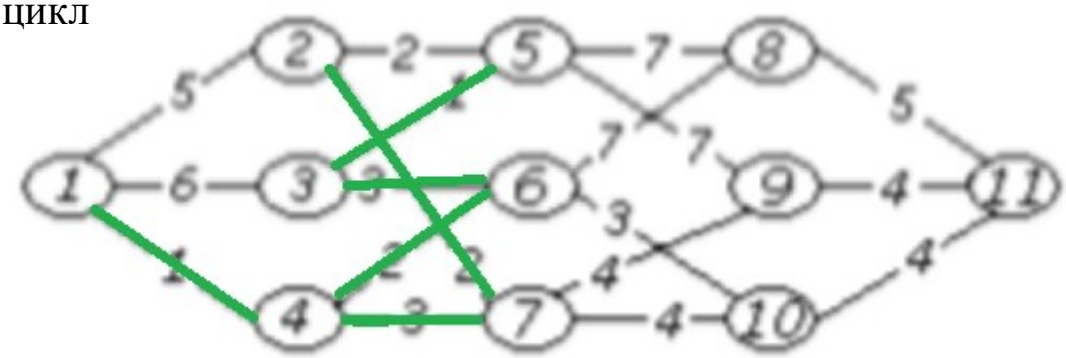


4. Аналогічними кроками доходимо до дерева

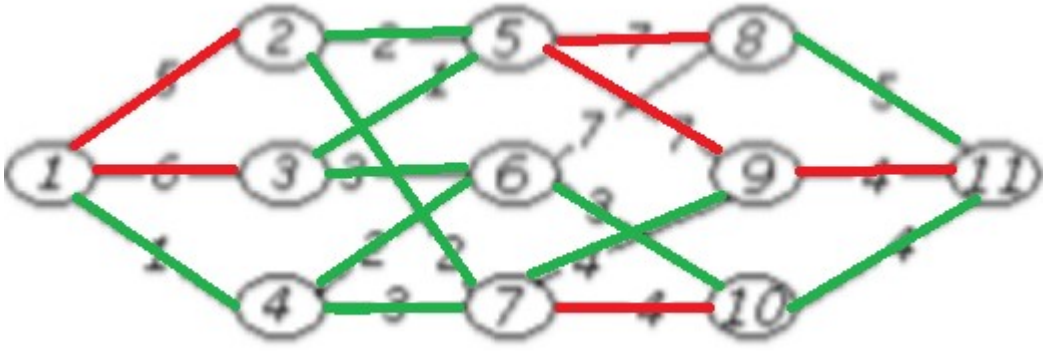


Алгоритм Краскала

Вибираємо ребра з найменшою довжиною які не утворюють цикл

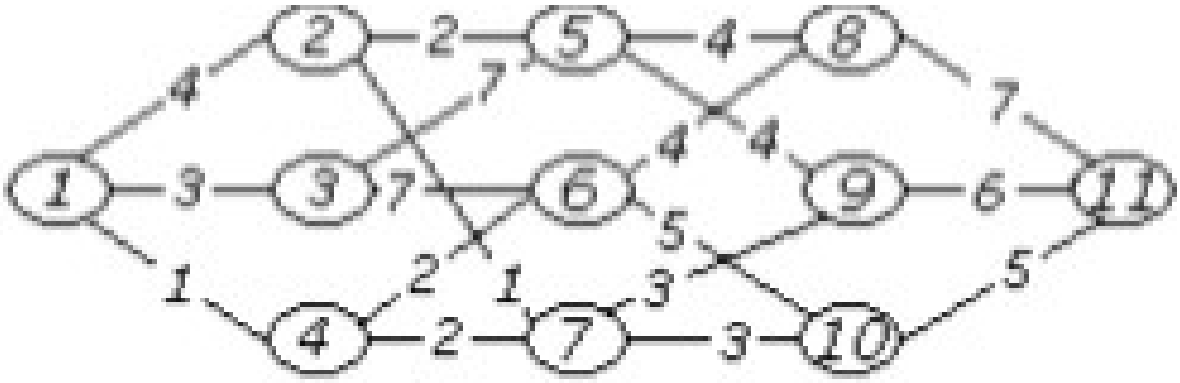


Ребра які утворюють цикл позначаємо червоним:

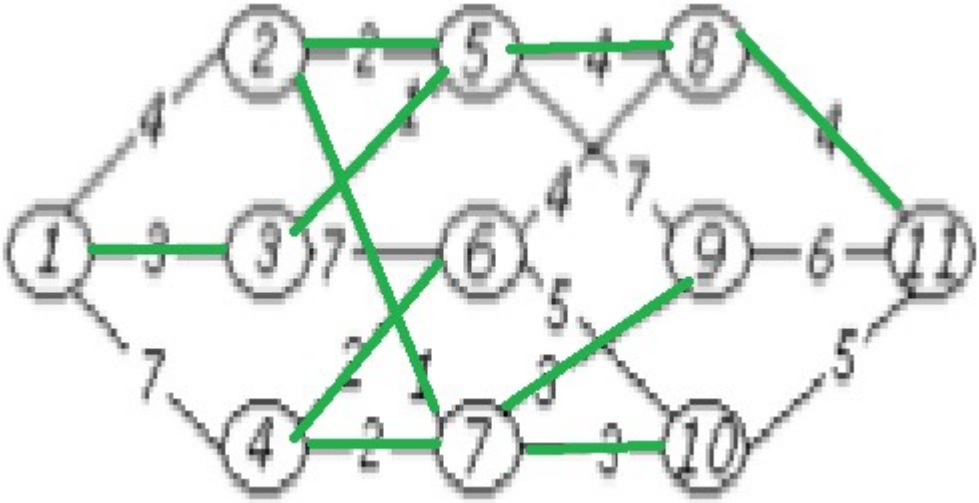


Завдання №2. Написати програму, яка реалізує алгоритм знаходження остового дерева мінімальної ваги згідно свого варіанту.

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



Програма має видати такий граф

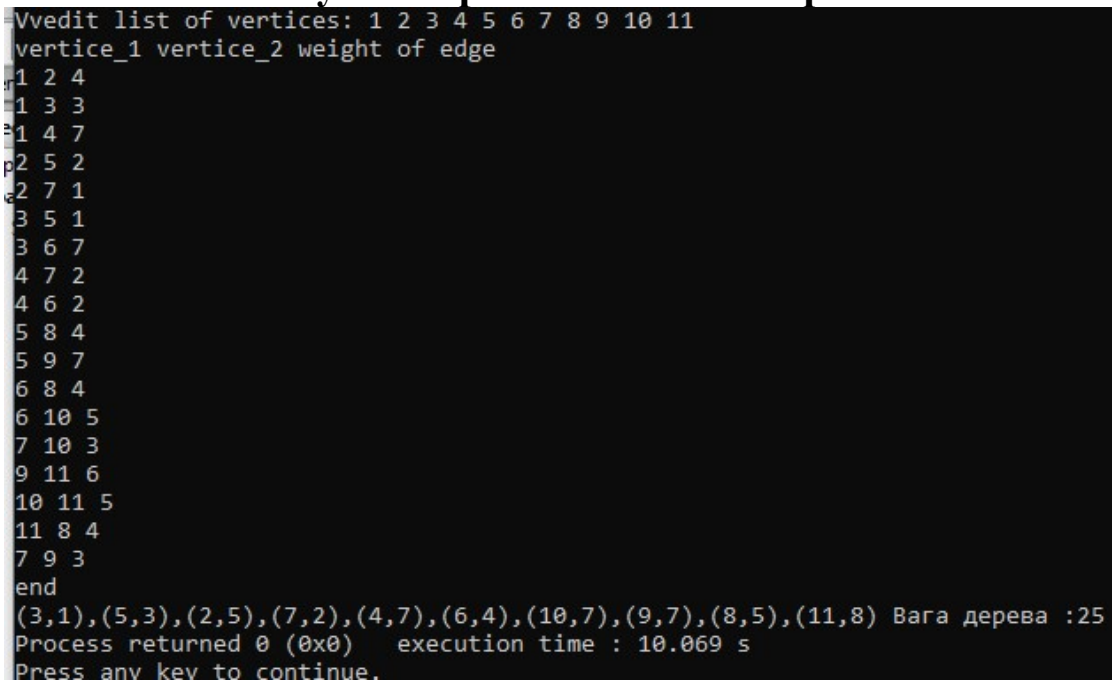


```
#include <iostream>
#include <set>2
#include <windows.h>
#include <Cstring>
using namespace std;
int consisting(int x,set <int> s){
int c = 0;
for(auto i : s){
    if (x == i)
        c = 1;
}
if (c ==1)
    return 1;
else
    return 0;
}
int main()
{
SetConsoleOutputCP(CP_UTF8);
set <int>  V;
set <int> seen;
char* s = new char[255];
cout<<"Vvedit list of vertices: ";
gets(s);
char* p;
p = strtok(s," ");
seen.insert(atoi(p));
cout<<"vertice_1 vertice_2 weight of edge\n";
while(p){
    V.insert(atoi(p));
    p = strtok(NULL," ");
```

```
set <int*> E;
gets(s);
while(strcmp(s,"end")){
    int* arr = new int[3];
    arr[0] = atoi(strtok(s," "));
    arr[1] = atoi(strtok(NULL , " "));
    arr[2] = atoi(strtok(NULL," "));
    if(arr[0] == 0||arr[1]==0||arr[2]==0){
        cout<<"Fuck";
        return 0;
    }

    E.insert(arr);
    gets(s);
}

set <int*> derevo;
int maximum = 0;
for(auto i : E){
    if (i[2] > maximum){
        maximum = i[2];
    }
}
int sum = 0;
int x = 0;
while(1){
    x++;
    int minimum = maximum;
    int new_vertice,pazyloi_vertice;
    if (sum!=0)
    cout<<",";
```

```
    for(auto i : E){
        if (consisting(i[0],seen)&&(consisting(i[1],seen)==0)){
            if (i[2] < minimum){
                minimum = i[2];
                new_vertice = i[1];
                pazyloi_vertice = i[0];
            }
        }
        if (consisting(i[1],seen)&&(consisting(i[0],seen) == 0)){
            if (i[2] <= minimum){
                minimum = i[2];
                new_vertice = i[0];
                pazyloi_vertice = i[1];
            }
        }

    }
    seen.insert(new_vertice);
    int* arr = new int[2];
    arr[0] = new_vertice;
    arr[1] = pazyloi_vertice;
    sum += minimum;
    derevo.insert(arr);
    cout<<"("<<arr[0]<<","<<arr[1]<<")";
    if (seen.size() == V.size())
        break;
    if (x > V.size()){
        system("cls");
        cout<<"Граф не звязний";
        sum = 0;
        break;
     }
}
if (sum !=0)
    cout<<" Вага дерева :"<<sum;
return 0;
}
```

Результат роботи компілятора:



Висновок: навчився працювати з графами використовуючи алгоритми Краскаля і Прима.