

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

Кафедра систем штучного інтелекту

Лабораторна робота №4

з дисципліни
«Дискретна математика»

Виконав:
студент групи КН-113
Омельюх Роман

Викладач: Мельникова Н.І.

Львів – 2019 р.

Лабораторна робота № 4.

Варіант №6

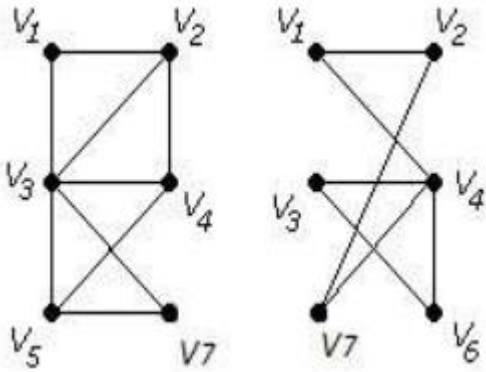
Тема: Основні операції над графами. Знаходження остова мінімальної ваги за алгоритмом Прима і Краскала

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

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

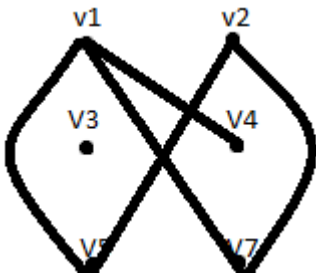
1. Виконати наступні операції над графами:

- 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 $G1$ ($G1 \setminus A$),
- 6) добуток графів.

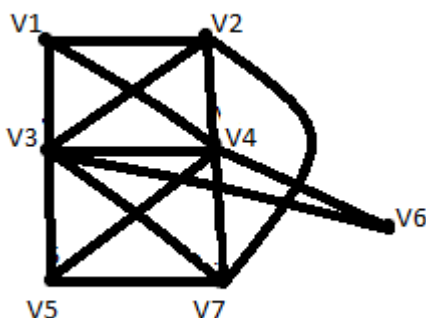


Розв'язки:

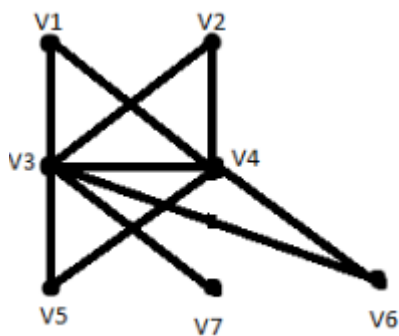
- 1) знайти доповнення до першого графу,



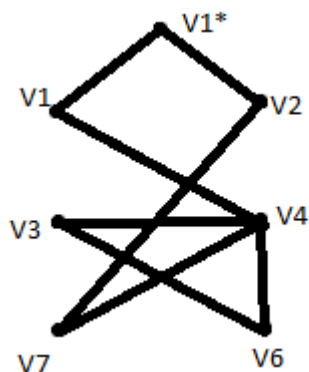
- 2) об'єднання графів,



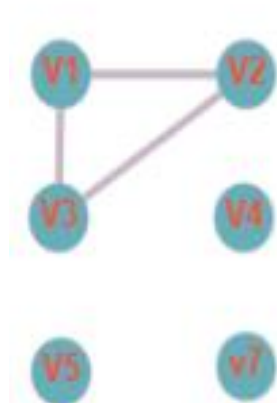
3) кільцеву суму $G1$ та $G2$ ($G1 \oplus G2$),



4) розщепити вершину у другому графі,

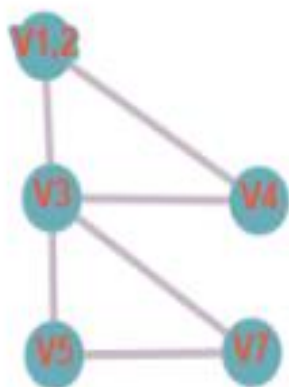


5) виділити підграф A , що складається з 3-х вершин в $G1$ і знайти стягнення A в $G1$ ($G1 \setminus A$),



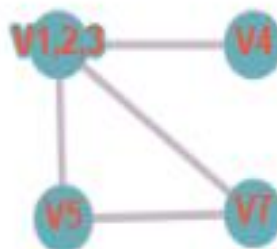
1)

Підграф A

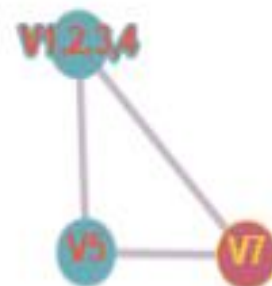


2)

Стягнення A в $G1$

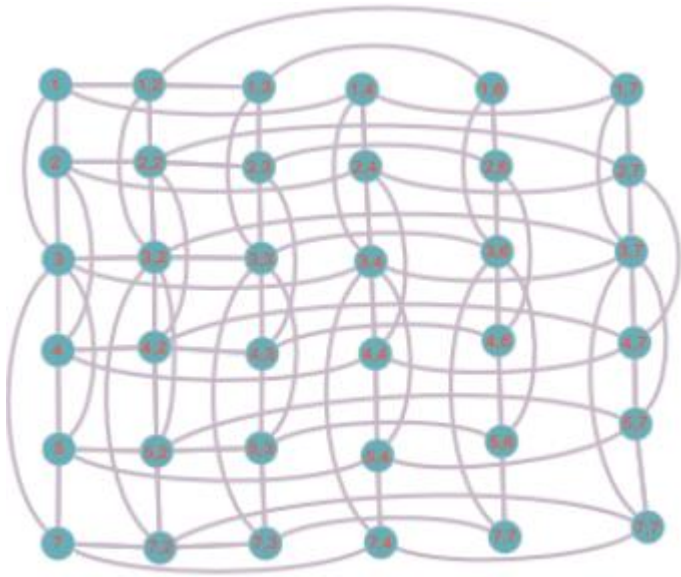


3)

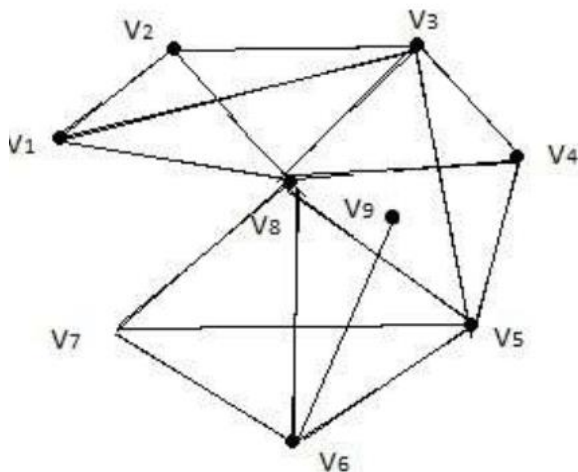


4)

б) добуток графів.



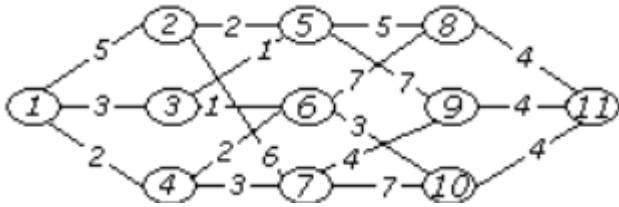
2. Знайти таблицю суміжності та діаметр графа.



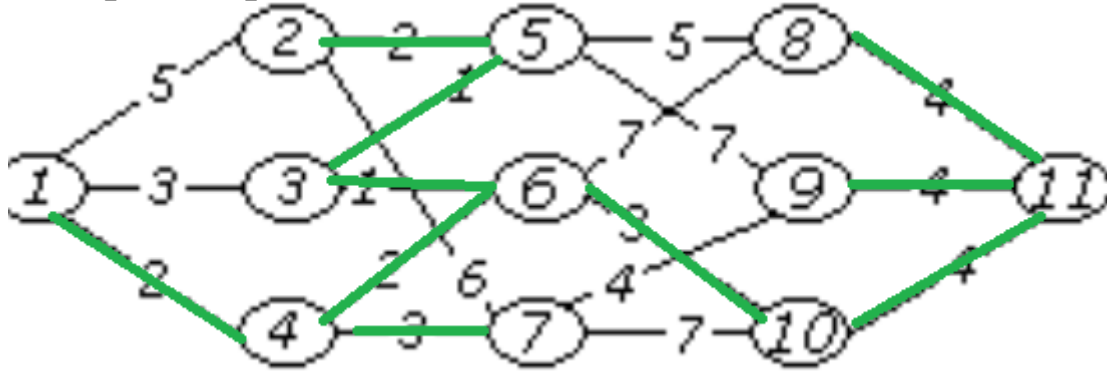
Діаметр: $d = 3(V1-V-V6-V9)$

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

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



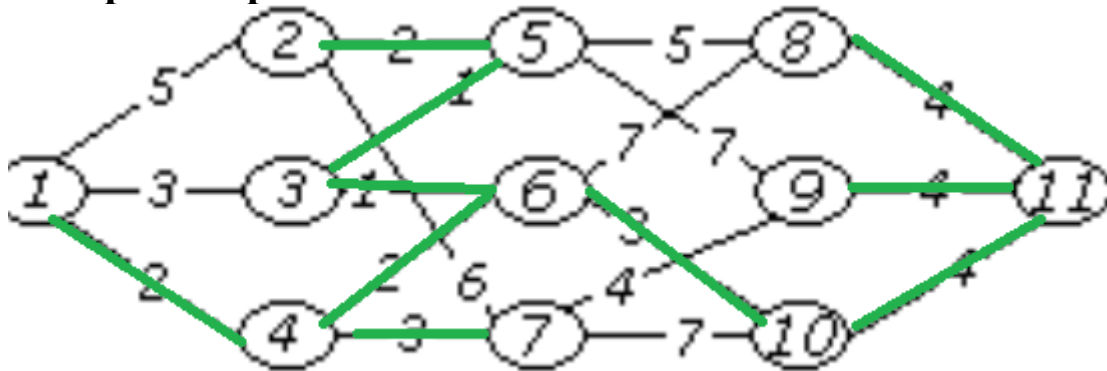
Алгоритм Краскала



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

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

Алгоритм Прима



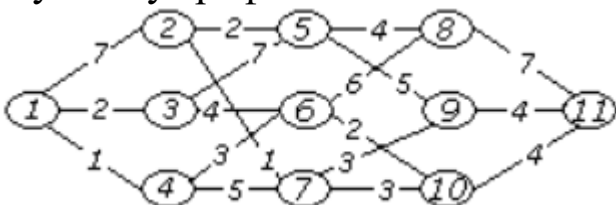
$V(t) = \{3, 5, 6, 4, 1, 2, 7, 10, 11, 8, 9\};$

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

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

Варіант № 6

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



```

1  #include <iostream>
2  #include <stdio.h>
3  using namespace std;
4  struct masl{
5      int vag;
6      int v1;
7      int v2;
8      bool in =false;
9  };
10 struct graf {
11     int arr[11][0];
12     int c=0;
13 };
14 int main(){
15     setlocale(LC_ALL, "Ukrainian");
16     int n=11,a=100,b=100,m=11;
17     int k;
18     graf inn[5];
19     masl mas[18];
20     int v1[18]={1,1,1,2,2,3,3,4,4,5,5,6,6,7,7,8,9,10};
21     int v2[18]={2,3,4,5,7,5,6,6,7,8,9,8,10,9,10,11,11,11};
22     int vag[18]={7,2,1,2,1,7,4,3,5,4,5,6,2,3,3,7,4,4};
23     for (int i=1;i<=18;i++){
24         mas[i].v1=v1[i-1];
25         mas[i].v2=v2[i-1];
26         mas[i].vag=vag[i-1];
27         for (int i=1;i<=18;i++){
28             for(int j=1;j<=18;j++){
29                 if(mas[j].vag>mas[i].vag){
30                     for(int j=1;j<=18;j++){
31                         if(mas[j].vag>mas[i].vag){
32                             swap(mas[i],mas[j]);
33                         }
34                     }
35                     for (int i=1;i<=18;i++){
36                         cout<<mas[i].v1<<" "<<mas[i].vag<<" "<<mas[i].v2<<endl;
37                     }
38                 }
39             }
40             int c=-1;
41             for (int i=0;i<18; i++) {
42                 for (int j=0;j<5; j++){
43                     for (int k=0;k<11; k++){
44                         if (mas[i].v1==inn[j].arr[k]) {a=j; goto point0; }}
45             point0;;
46             for (int j=0;j<5; j++)
47             {
48                 for (int k=0;k<11; k++)
49                 {
50                     if (mas[i].v2==inn[j].arr[k]){b=j; goto point1; }
51                 }
52             }
53             point1;;
54             if (a!=b && a==100) {inn[b].arr[inn[b].c]=mas[i].v1; inn[b].c++; }
55             if (a!=b && b==100) {inn[a].arr[inn[a].c]=mas[i].v2; inn[a].c++; }
56             if (a!=b && a!=100 && b!=100) {
57                 if (a<b) {
58                     for (int l=0;l<inn[b].c; l++)
59                     {

```

```

55     for (int l=0;l<inn[b].c; l++)
56     {
57         inn[a].arr[inn[a].c+l]=inn[b].arr[l];
58         inn[b].arr[l] = 0;
59     }
60     inn[a].c+=inn[b].c;
61     inn[b].c =0;
62 }
63 if (b<a) {
64     for (int l=0;l<inn[a].c; l++)
65     {
66         inn[b].arr[inn[b].c+l]=inn[a].arr[l];
67         inn[a].arr[l] ;
68     }
69     inn[b].c+=inn[a].c;
70     inn[a].c=0;
71 }
72 }
73 if(a==100 && b==100) {c++;inn[c].arr[inn[c].c]=mas[i].v1; inn[c].arr[inn[c].c + 1]=mas[i].v2; inn[c].c += 2; }
74
75 mas[i].in = true;
76 if (a==b && a!=100) {mas[i].in=false;} a=100;b=100;
77 }
78 cout << "Остове дерево мінімальної ваги, ми повинні включити в нього такі ребра: " << endl;
79 int sum= 0;
80 for (int i=1;i<=m; i++)
81 {
82     if (mas[i].in==true) {cout<<mas[i].v1<< " " <<mas[i].v2<<" "<<mas[i].v2<<endl;sum+=mas[i].vag; }
83 }

```

```

79     int sum= 0;
80     for (int i=1;i<=m; i++)
81     {
82         if (mas[i].in==true) {cout<<mas[i].v1<< " " <<mas[i].v2<<" "<<mas[i].v2<<endl;sum+=mas[i].vag; }
83     }
84     cout << "Остове дерево мінімальної ваги для даного графа: " << sum;
85     return 0;}

```

Результати:

```

1 2 3
6 2 10
4 3 6
7 3 9
7 3 10
5 4 8
3 4 6
9 4 11
10 4 11
5 5 9
4 5 7
6 6 8
8 7 11
3 7 5
1 7 2
остове дерево мінімальної ваги, ми повинні включити в нього такі ребра:
1 1 4
2 1 7
2 2 5
1 2 3
6 2 10
4 3 6
7 3 9
7 3 10
5 4 8
9 4 11
Остове дерево мінімальної ваги для даного графа: 25
Process returned 0 (0x0)   execution time : 0.153 s
Press any key to continue.

```

Висновок: на цій лабораторній ми набули практичних вмінь та навичок з використанням алгоритмів Прима і Крaskала.