

**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
Коваленко Антон
Викладач:
Мельникова Н.І.

Львів – 2019 р.

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.

Додаток 1

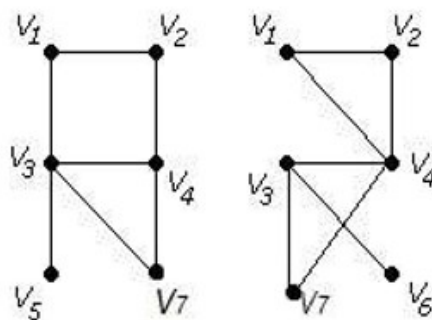
Варіант 13

1.Завдання:

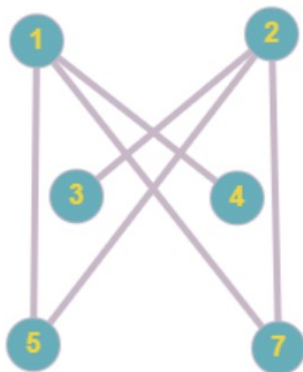
Виконати наступні операції над графами:

- 1) знайти доповнення до першого графу;
- 2) об'єднання графів;
- 3) кільцеву суму $G1$ та $G2$;
- 4) розщепити вершину у другому графі;
- 5) виділити підграф A , що складається з 3-х вершин в $G1$ і знайти стягнення A в $G1$;
- 6) добуток графів;

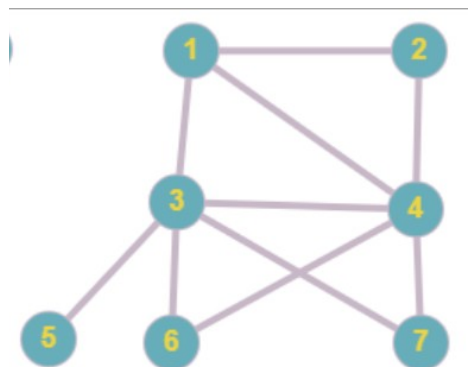
13



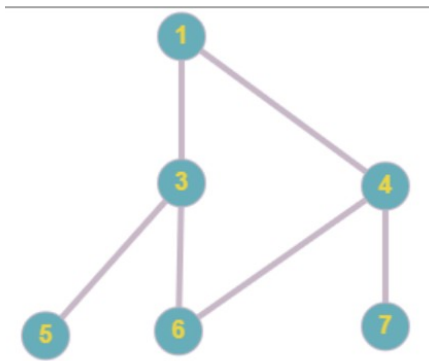
1)



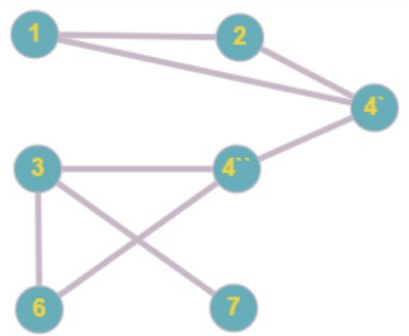
2)



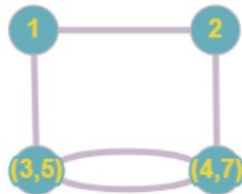
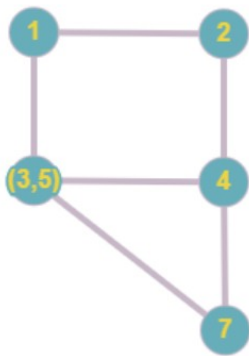
3)



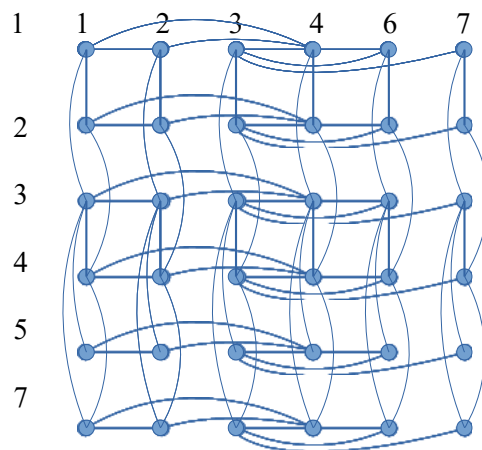
4)



5)

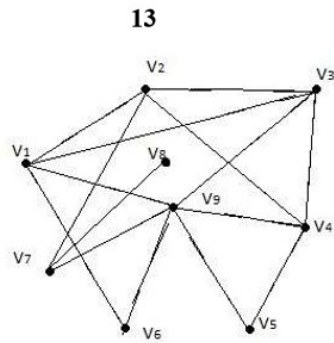


6)



2.Завдання

Знайти таблицю суміжності та діаметр графа



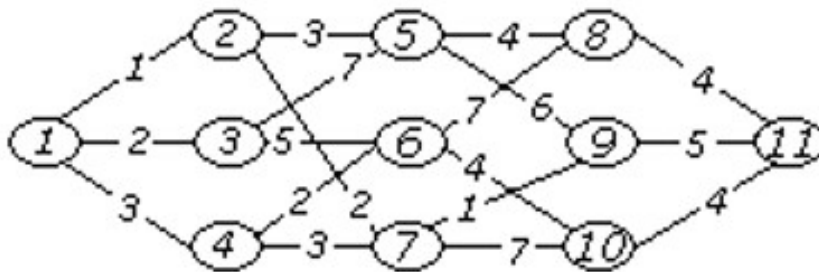
	V1	V2	V3	V4	V5	V6	V7	V8	V9
V1	0	1	1	0	0	1	0	0	1
V2	1	0	1	1	0	0	1	0	0
V3	1	1	0	1	0	0	0	0	1
V4	0	1	1	0	1	0	0	0	1
V5	0	0	0	1	0	0	0	0	1
V6	1	0	0	0	0	0	0	0	1
V7	0	1	0	0	0	0	0	1	1
V8	0	0	0	0	0	0	1	0	0
V9	1	0	1	1	1	1	1	0	0

Діаметр графа = 3.

3.Завдання

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

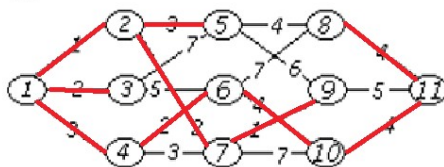
13



Прима: 9-7, 7-2, 2-1, 1-3, 1-4, 4-6, 2-5, 6-10, 10-11, 11-8.

Краскала: 1-2, 7-9, 7-2, 1-3, 4-6, 1-4, 2-5, 6-10, 10-11, 8-11.

13

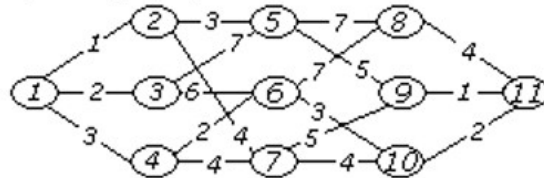


Додаток 2

Написати програму, яка реалізує алгоритм знаходження мінімального остового дерева мінімальної ваги згідно свого варіанту.

Варіант № 13

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>
#define m 11
int main()
{
    setlocale(LC_CTYPE,"ukr");
    int graph[m][m]={
        {0,1,2,3,0,0,0,0,0,0,0},
        {1,0,0,0,3,0,4,0,0,0,0},
        {2,0,0,0,7,6,0,0,0,0,0},
        {3,0,0,0,0,2,4,0,0,0,0},
        {0,3,7,0,0,0,0,7,5,0,0},
        {0,0,6,2,0,0,0,7,0,3,0},
        {0,0,0,4,0,0,0,0,5,4,0},
        {0,0,0,0,0,7,0,0,0,0,4},
        {0,0,0,0,5,0,5,0,0,0,1},
        {0,0,0,0,0,3,4,0,0,0,2},
        {0,0,0,0,0,0,0,4,1,2,0}
    };
    int i,j,p=0,q=0;
    printf("Матриця суміжності:\n\t");
    for(i=0;i<m;i++){
        for(j=0;j<m;j++){
            printf("%d ",graph[i][j]);
        }
        printf("\n\t");
    }
    printf("\nРедра, що входять в мінімальне остове дерево:\n");
    int visit[m]={0};
    int arr[m]={1,2,3,4,5,6,7,8,9,10,11};
    int min;
```

```

int flag=0;
for(i=0;i<m;i++){
for(j=0;j<m;j++){
if(flag==0&&graph[i][j]!=0){
flag=1;
p=i;
q=j;
min=graph[p][q];
}
else if(flag==1&&graph[i][j]<min&&graph[i][j]!=0){
p=i;
q=j;
min=graph[i][j];
}
}
}
visit[p]=1;
visit[q]=1;
int flag1=0;
int p1,q1,min1,qq=0;
printf("%d-->%d",arr[p],arr[q]);
do{
for(i=0;i<m;i++){
for(j=0;j<m;j++){
if(visit[i]==1 && visit[j]==0 && graph[i][j]!=0){
if(flag1==0){
flag1 = 1;
p1=i;
q1=j;
min1=graph[i][j];
}else if(flag1==1&&graph[i][j]<min1){
p1=i;
q1=j;
min1=graph[i][j];
}
}
}
}
}
visit[q1]=1;
flag1=0;
printf("\n%d-->%d",arr[p1],arr[q1]);
qq++;
}
while(qq<m-2);
return 0;
}

```

Результати виконання:

Матриця суміжності:

```
0 1 2 3 0 0 0 0 0 0 0
1 0 0 0 3 0 4 0 0 0 0
2 0 0 0 7 6 0 0 0 0 0
3 0 0 0 0 2 4 0 0 0 0
0 3 7 0 0 0 0 7 5 0 0
0 0 6 2 0 0 0 7 0 3 0
0 0 0 4 0 0 0 0 5 4 0
0 0 0 0 0 7 0 0 0 0 4
0 0 0 0 5 0 5 0 0 0 1
0 0 0 0 0 3 4 0 0 0 2
0 0 0 0 0 0 0 4 1 2 0
```

Ребра, що входять в мінімальне остове дерево:

```
1-->2
1-->3
1-->4
4-->6
2-->5
6-->10
10-->11
11-->9
2-->7
11-->8
```

Висновок

На лабораторній роботі я ознайомився з основним поняттям графів, основними операціями над графами та алгоритмом пошуку мінімального остового дерева Прима, Краскала.