

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра систем штучного інтелекту

Лабораторна робота №4

з дисципліни

«Дискретна математика»

Виконала:

студентка групи КН-112

Максимець Віра

Перевірила:

Мельникова Н.І.

Львів-2019

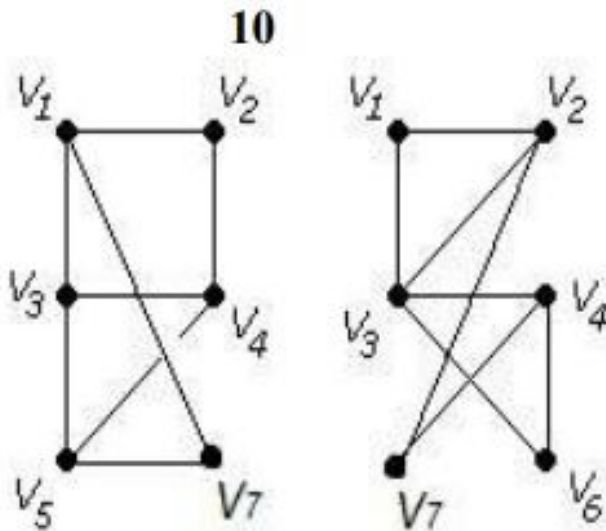
Тема: Основні операції над графами. Знаходження остова мінімальної ваги за алгоритмом Пріма-Краскала

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

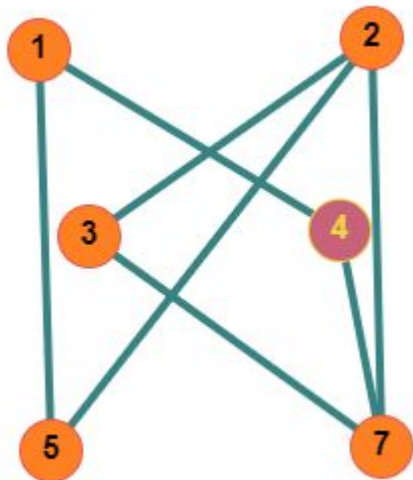
Варіант №10

1. Виконати наступні операції над графами:

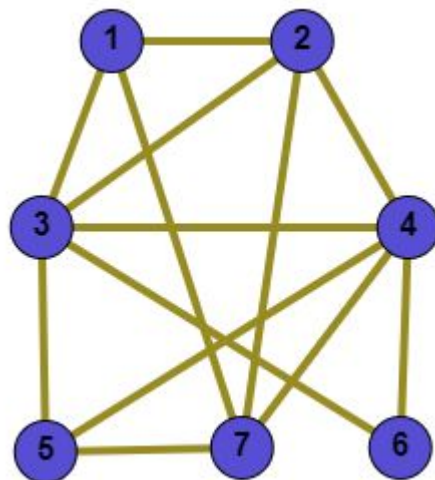
- 1) знайти доповнення до першого графу,
- 2) об'єднання графів,
- 3) кільцеву суму $G1$ та $G2$ ($G1+G2$),
- 4) розщепити вершину у другому графі,
- 5) виділити підграф A , що складається з 3-х вершин в $G1$ і знайти стягнення A в $G1$ ($G1 \setminus A$),
- 6) добуток графів.

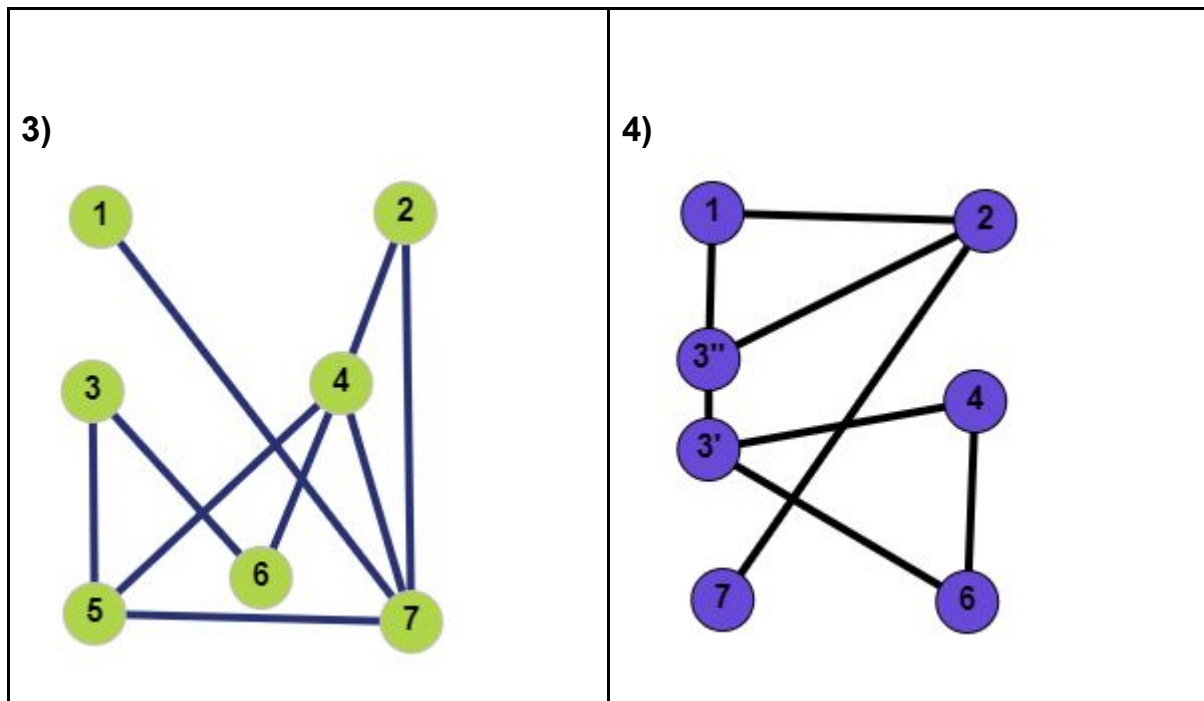


1)

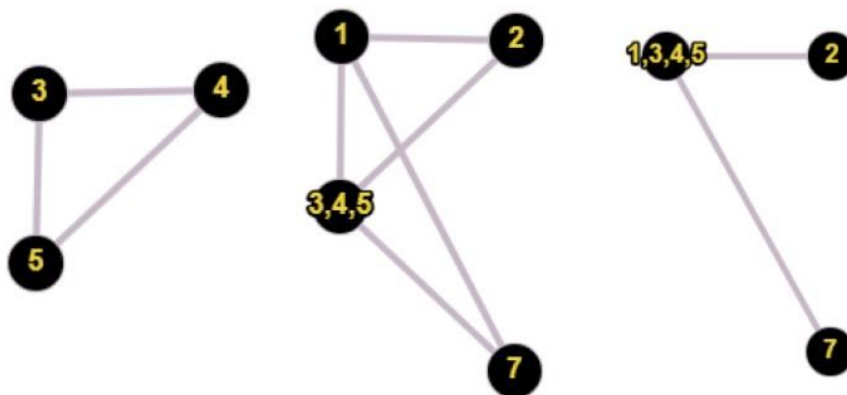


2)



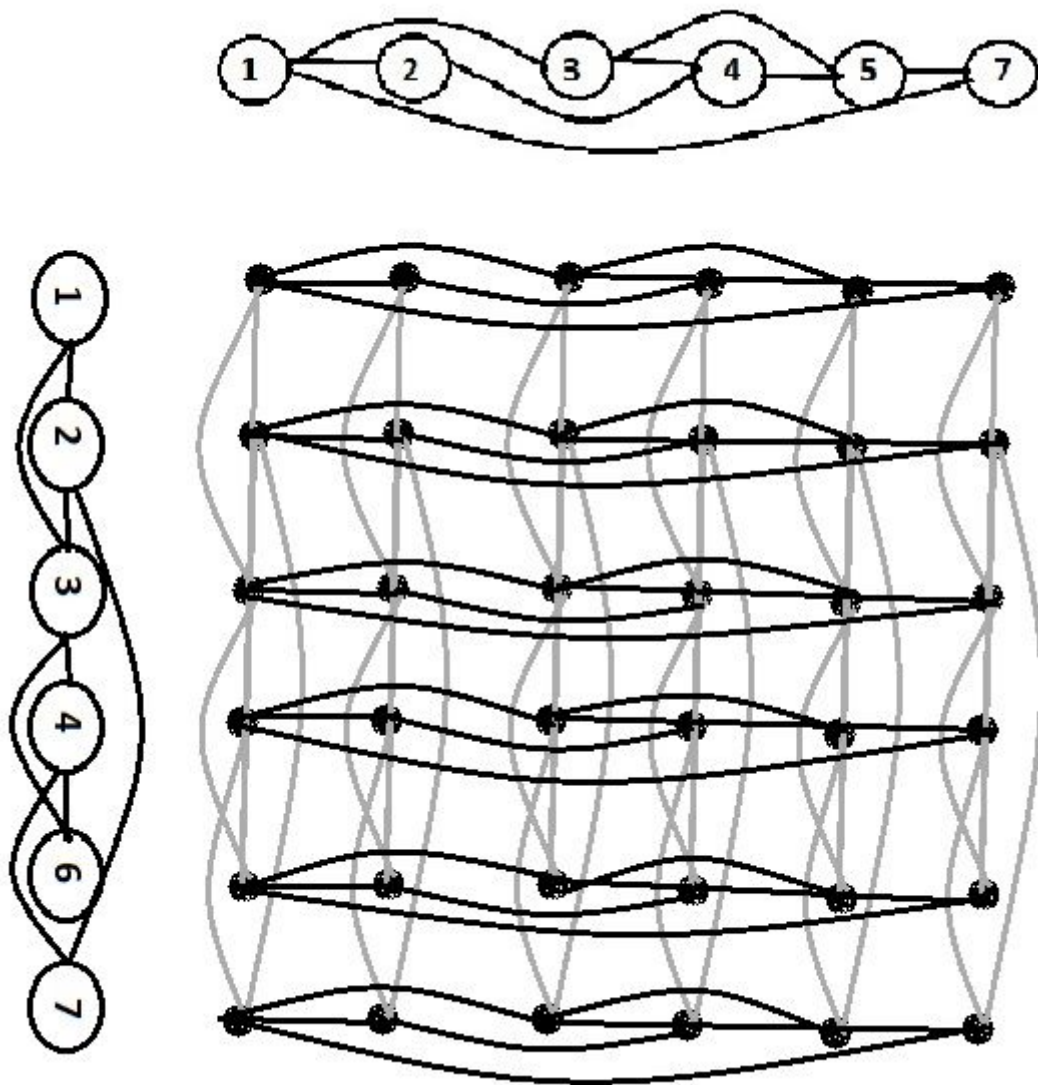


5) виділити підграф A, що складається з 3-х вершин в G_1 - і знайти стягнення A в G_1 ($G_1 \setminus A$)



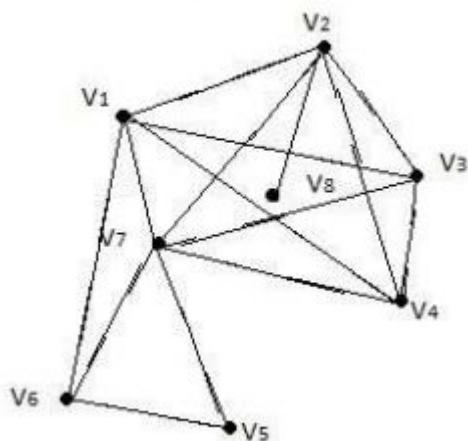
6) добуток графів

Щоб знайти добуток графів, кожен з них перемістимо так, щоб вершини утворювали лінію



2.Знайти таблицю суміжності та діаметр графа.

10

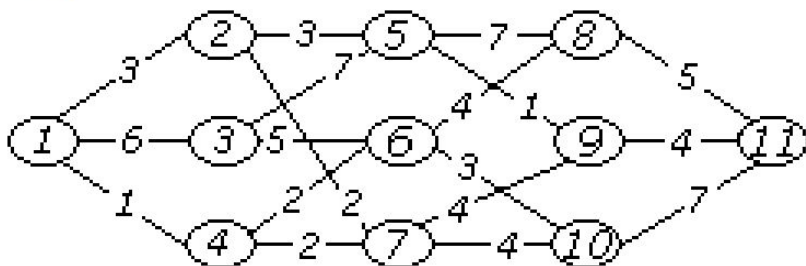


V	1	2	3	4	5	6	7	8
1	-	1	1	1	0	1	1	0
2	1	-	1	1	0	0	1	1
3	1	1	-	1	0	0	1	0
4	1	1	1	-	0	0	1	0
5	0	0	0	0	-	1	1	0
6	1	0	0	0	1	-	1	0
7	1	1	1	1	1	1	-	0
8	0	1	0	0	0	0	0	-

Діаметр - 4 (5-8)

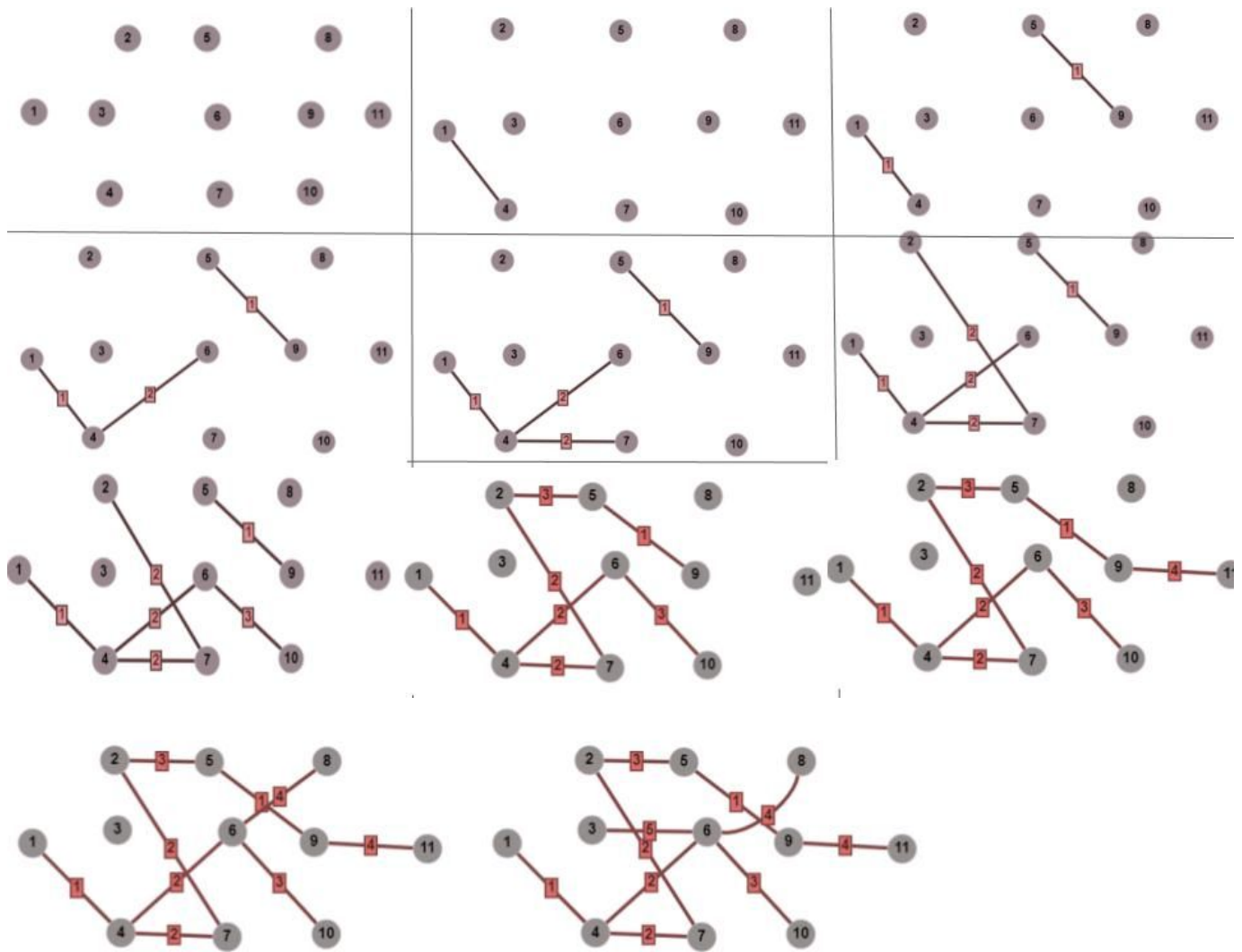
3. Знайти двома методами (Краскала і Прима) мінімальне остове дерево графа.

10



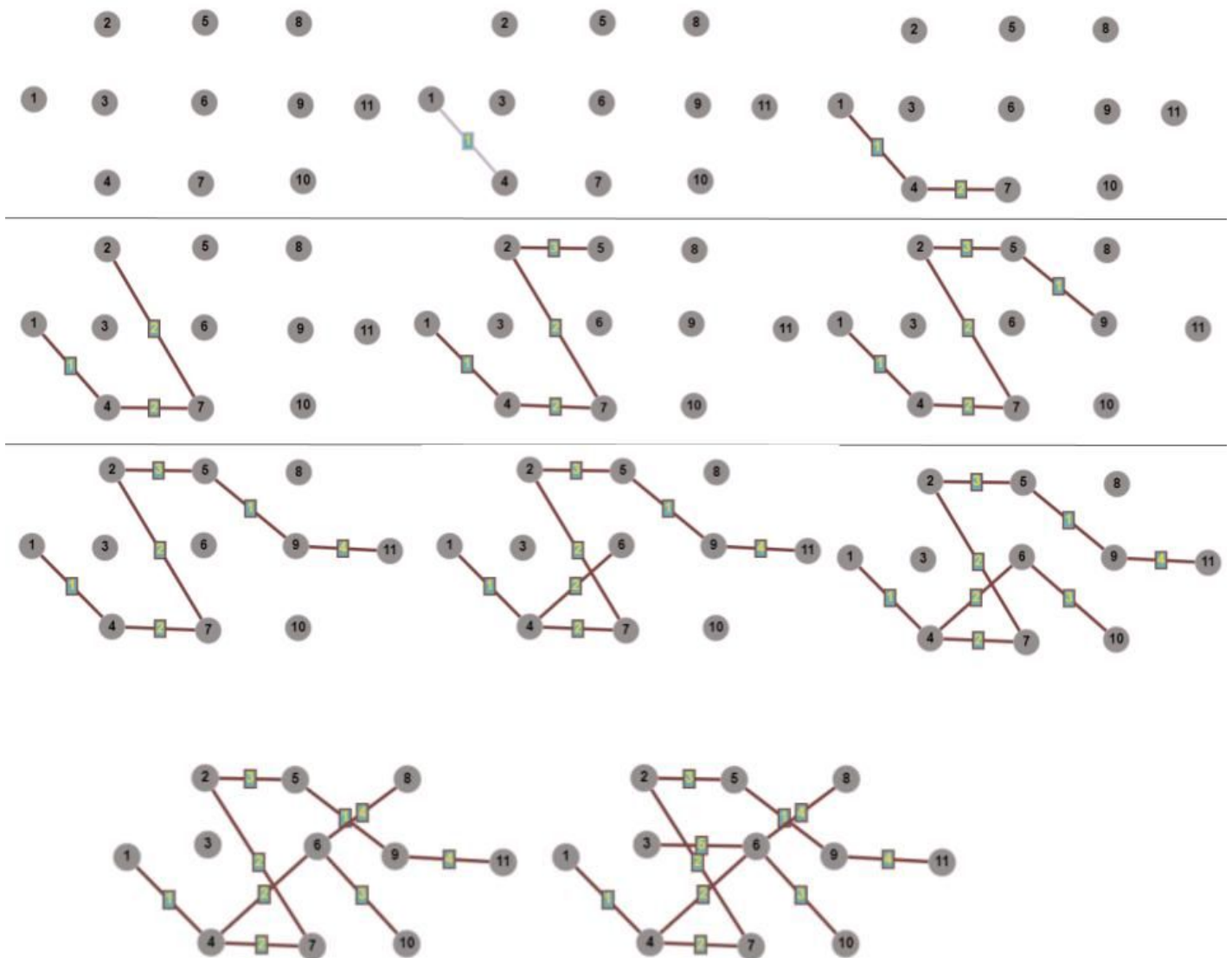
Метод Краскала

1. V1+V4
2. V5+V9
3. V4+V6
4. V4+V7
5. V7+V2
6. V6+V10
7. V2+V5
8. V9+V11
9. V6+V8
10. V6+V3



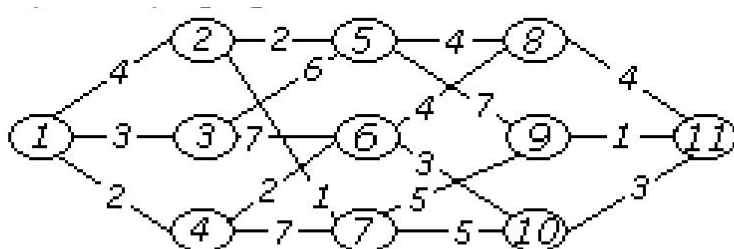
Метод Прима

- 1) V_1+V_4
- 2) V_4+V_7
- 3) V_7+V_2
- 4) V_2+V_5
- 5) V_5+V_9
- 6) V_9+V_{11}
- 7) V_4+V_6
- 8) V_6+V_{10}
- 9) V_6+V_8
- 10) V_6+V_3



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

За алгоритмом Краскала знайти мінімальне остове дерево графа.



Етапи розв'язання задачі виводити на екран. Протестувати розроблену програму на графі.

Код програми

```
1  #include <iostream>
2
3  using namespace std;
4
5  void Probe(int v, int AM[][11], int Values[], int Lines[]);
6
7  int main()
8  {
9      int AM[11][11] =
10     {
11         {99, 4, 3, 2, 99, 99, 99, 99, 99, 99, 99},
12         {4, 99, 99, 99, 2, 99, 1, 99, 99, 99, 99},
13         {3, 99, 99, 99, 6, 7, 99, 99, 99, 99, 99},
14         {2, 99, 99, 99, 99, 2, 7, 9, 99, 99, 99},
15         {99, 2, 6, 99, 99, 99, 99, 4, 7, 99, 99},
16         {99, 99, 7, 2, 99, 99, 99, 4, 99, 3, 99},
17         {99, 1, 99, 7, 99, 9, 99, 99, 5, 5, 99},
18         {99, 99, 99, 99, 4, 4, 99, 99, 99, 99, 4},
19         {99, 99, 99, 99, 7, 99, 5, 99, 99, 99, 1},
20         {99, 99, 99, 99, 99, 3, 5, 99, 99, 99, 3},
21         {99, 99, 99, 99, 99, 99, 99, 99, 4, 1, 3, 99}
22     };
23
24     //веса ребер
25     int Values[]={1, 2, 3, 4, 5, 6, 7}; // "v"
26     int value = (sizeof(Values))/4;
27
28     //массив для записи пройденных ребер
29     int Lines[11];
30     //занулить
31     for (int i=0; i<value; i++)
```

```
31     {
32         Lines[i]=0;
33     }
34     //вызов функции
35     for (int weight=0; weight<value; weight++)
36     {
37         Probe(weight, AM, Values, Lines);
38     }
39     return 0;
40 }
41 void Probe(int v, int AM[][11], int Values[], int Lines[])
42 {
43     int counter1 = 0;
44     int counter2 = 0;
45     bool flag1, flag2;
```



```

46
47     for (int i=0;i<11;i++)
48     {
49         for (int j=0;j<11;j++)
50         {
51             if (AM[i][j]==Values[v])
52             {
53                 for (int x=0;x<11;x++)
54                 {
55                     if (Lines[x]!=i)    //якщо нема ще такого ребра
56                     {
57                         counter1++;
58                     }
59                     if (Lines[x]!=j)    //якщо нема ще такого ребра
60                     {

```

```

61                         counter2++;
62                     }
63                 }
64                 if (counter1==11)
65                 {
66                     Lines[i]=i;
67                     flag1 = true;
68                 }
69                 if (counter2==11)
70                 {
71                     Lines[j]=j;
72                     flag2 = true;
73                 }
74                 if ((flag1==false)&&(flag2==false))
75                 { }
76                 else {
77                     cout << "\n Connection between: {" << Lines[i]+1 << " " << Lines[j]+1 << " }\n";
78                     cout << "Line weight: " << v+1 << endl;;
79                 }
80             }
81         }
82         counter1=0;
83         counter2=0;
84         flag1 = false;
85         flag2 = false;
86     }
87 }
88 }
89

```

```
C:\Users\User_2\Desktop\Dis444\bin\Debug\Dis444.exe

Connection between: <2;7>
Line weight: 1

Connection between: <9;11>
Line weight: 1

Connection between: <1;4>
Line weight: 2

Connection between: <2;5>
Line weight: 2

Connection between: <4;6>
Line weight: 2

Connection between: <1;3>
Line weight: 3

Connection between: <6;10>
Line weight: 3

Connection between: <5;8>
Line weight: 4

Process returned 0 (0x0)   execution time : 0.113 s
Press any key to continue.
```

Висновок

Я набула практичних вмінь та навичок з використання алгоритмів Прима і Краскала.