

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ «ЛЬВІВСЬКА
ПОЛІТЕХНІКА»**

Кафедра систем штучного інтелекту

Лабораторна робота №4
з дисципліни
«Дискретна математика»

Виконав:
студент групи КН-112
Лужняк Олексій
Перевірила:
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.

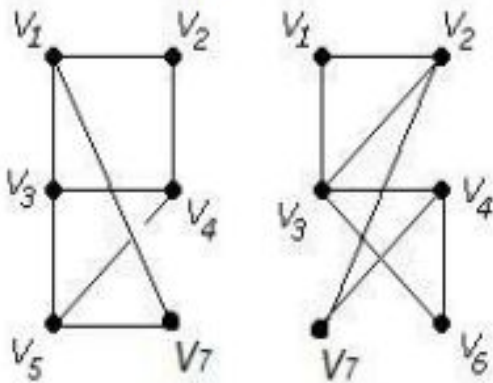
Варіант №10

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

1. Виконати наступні операції над графами:

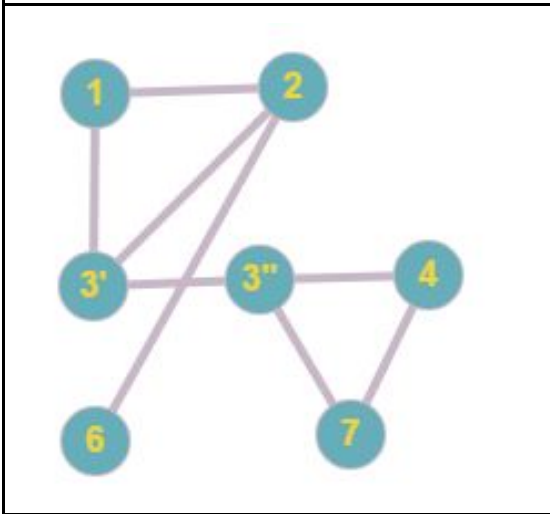
- 1) знайти доповнення до першого графу,
- 2) об'єднання графів,
- 3) кільцеву суму $G1$ та $G2$ ($G1+G2$),
- 4) розщепити вершину у другому графі,
- 5) виділити підграф A , що складається з 3-х вершин в $G1$ і знайти стягнення A в $G1$ ($G1 \setminus A$), 6) добуток графів.

10

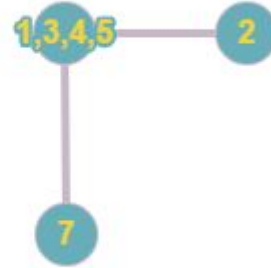
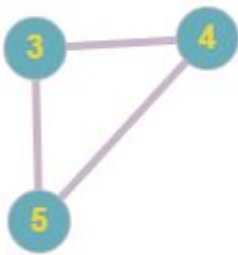


1)Доповнення графу	2)Об'єднання графів	3)Кільцева сума $G1$ та $G2$

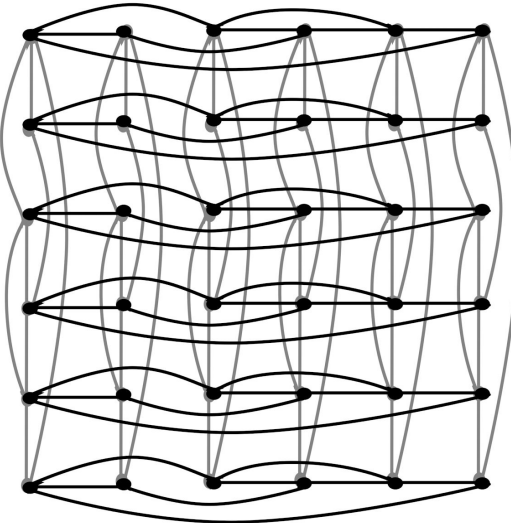
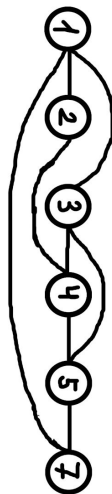
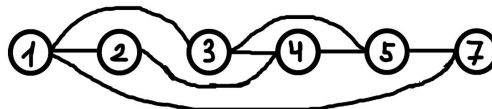
4) Розчепити вершину у другому графі



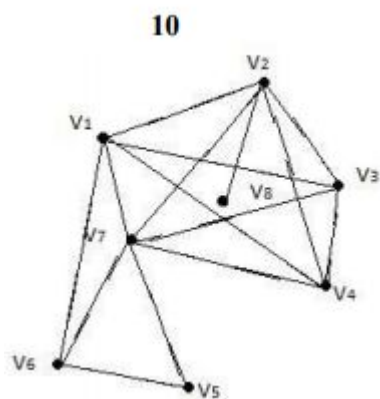
5) виділити підграф A, що складається з 3-х вершин в G_1 - і знайти стягнення A в G_1 ($G_1 \setminus A$)



6) Добуток графів. Щоб знайти добуток графів, кожен з них перемістимо так, щоб вершини утворювали лінію.



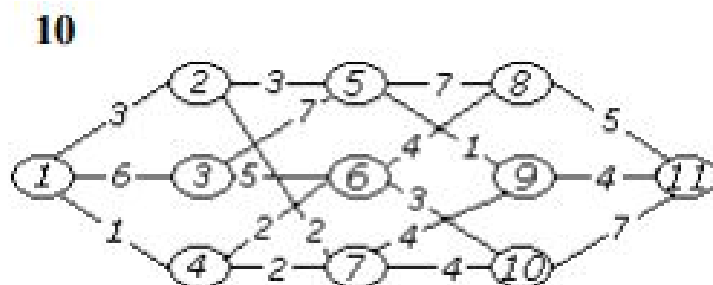
2. Знайти таблицю суміжності та діаметр графа.



	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	-

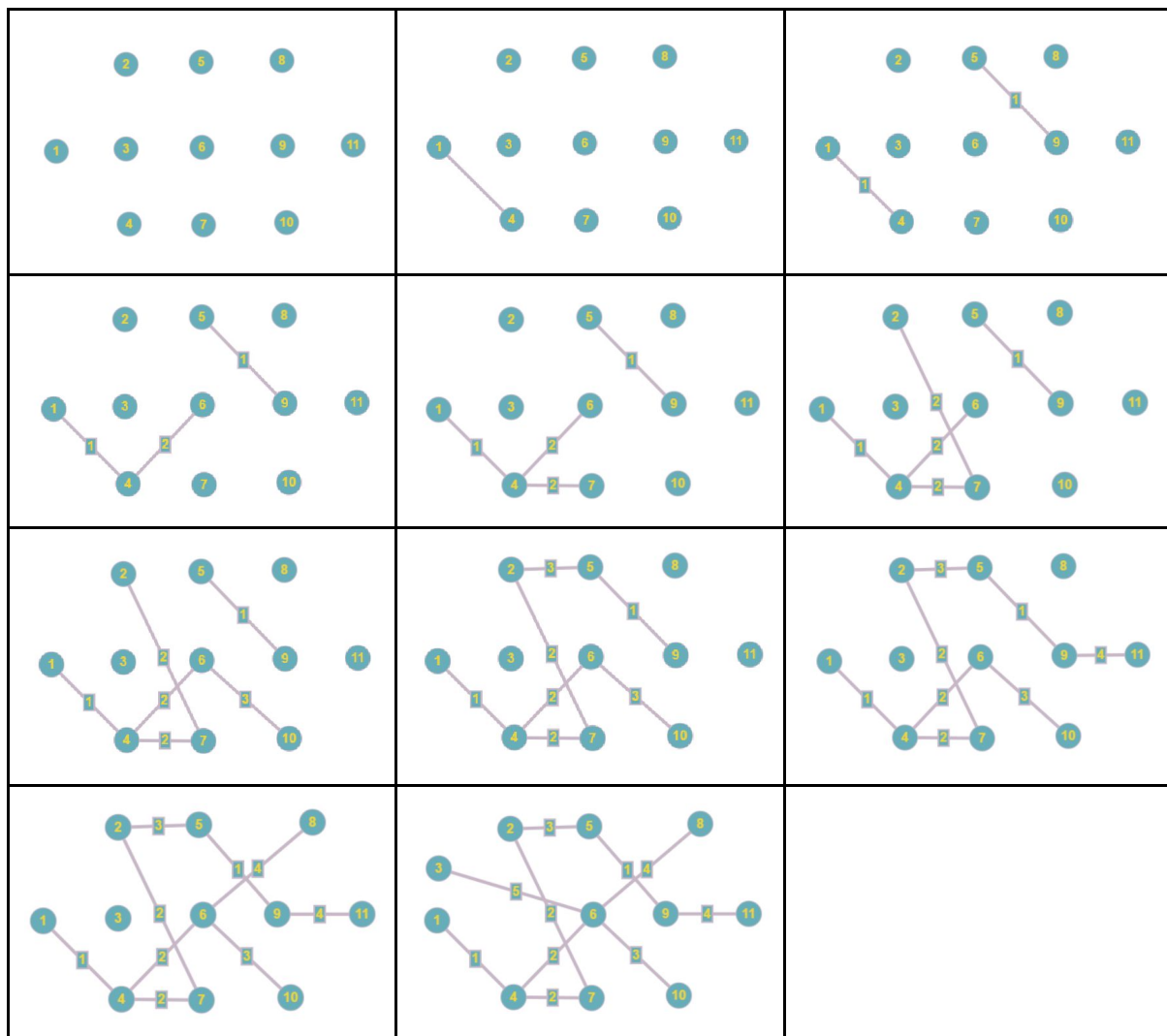
Діаметр - 4 (5-8)

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



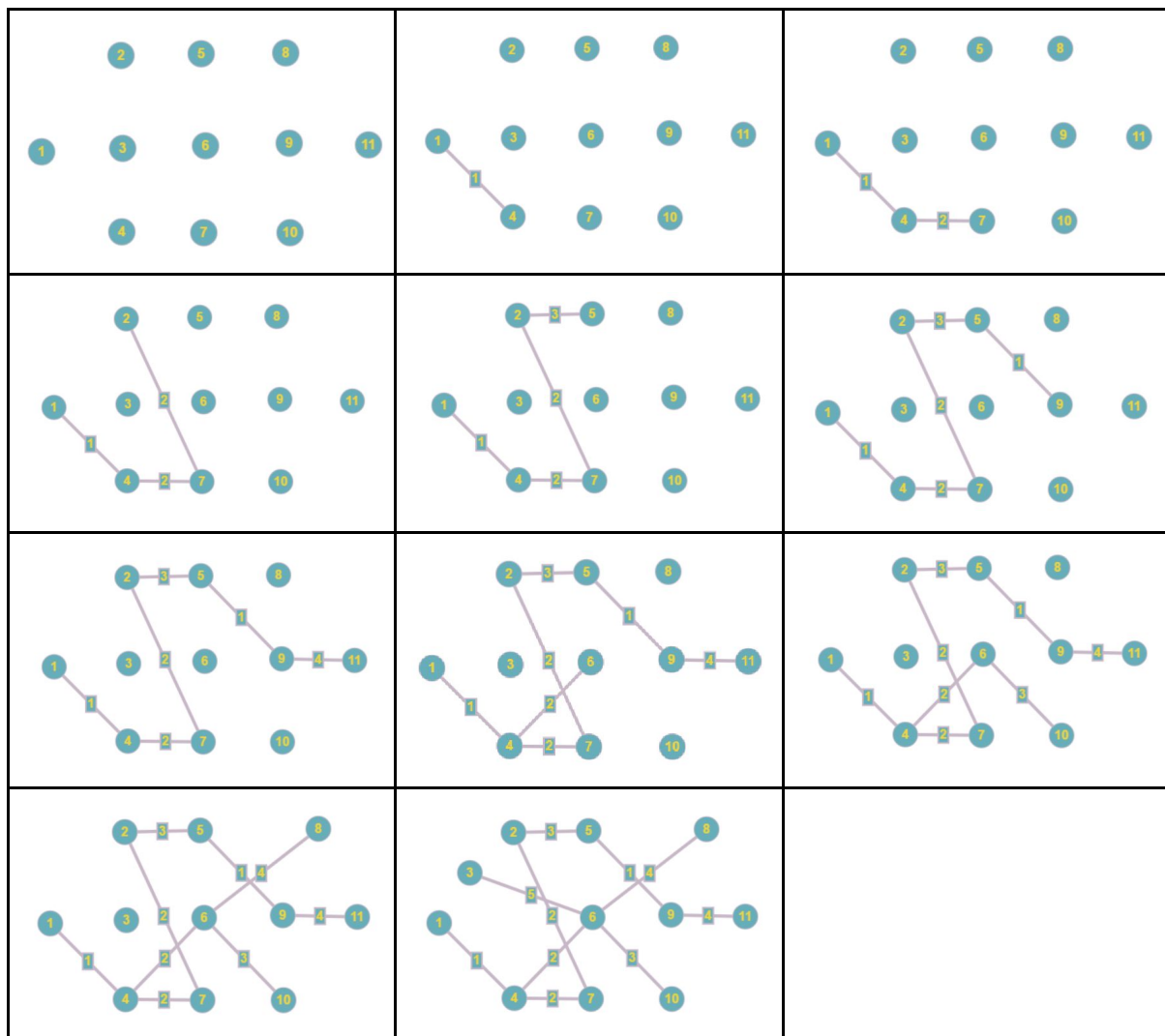
Метод Краскала

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



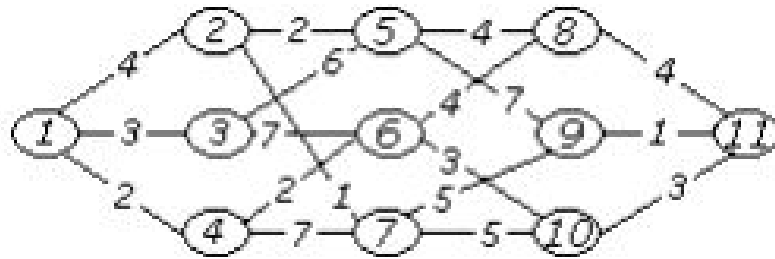
Метод Прима

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



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

варіанту. За алгоритмом Крaskала знайти мінімальне остове дерево графа.



```

1  #include <iostream>
2  using namespace std;
3
4  void Probe(int v, int AM[][11], int Values[], int Lines[]);
5
6  int main()
7  {
8      int AM[11][11] =
9      { {99,4,3,2,99,99,99,99,99,99,99},
10      {4,99,99,99,2,99,1,99,99,99,99},
11      {3,99,99,99,6,7,99,99,99,99,99},
12      {2,99,99,99,99,2,7,9,99,99,99},
13      {99,2,6,99,99,99,99,4,7,99,99},
14      {99,99,7,2,99,99,99,4,99,3,99},
15      {99,1,99,7,99,9,99,99,5,5,99},
16      {99,99,99,99,4,4,99,99,99,99,4},
17      {99,99,99,99,7,99,5,99,99,99,1},
18      {99,99,99,99,99,3,5,99,99,99,3},
19      {99,99,99,99,99,99,99,4,1,3,99}
20  };
21
22  //вара ребер
23  int Values[] = { 1,2,3,4,5,6,7 }; // "v"
24  int Value = (sizeof(Values)) / 4;
25
26  // масив для запису пройдених ребер
27  int Lines[11];
28  //занулити
29  for (int i = 0; i < Value; i++)
30  {
31      Lines[i] = 0;
32  }

```

```

33 //виклик функції
34 for (int weight = 0; weight < Value; weight++)
35 {
36     Probe(weight, AM, Values, Lines);
37 }
38 return 0;
39 }
40 void Probe(int v, int AM[][11], int Values[], int Lines[])
41 {
42     int counter1 = 0;
43     int counter2 = 0;
44     bool flag1, flag2;
45
46     for (int i = 0; i < 11; i++)
47     {
48         for (int j = 0; j < 11; j++)
49         {
50             if (AM[i][j]==Values[v])
51             {
52                 for (int x = 0; x < 11; x++)
53                 {
54                     if (Lines[x]!=i)
55                     {
56                         counter1++;
57                     }
58                     if (Lines[x] != j)
59                     {
60                         counter2++;
61                     }

```

```

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

```



```
Kruskal method  
  
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
```

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