

**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
Бондар А.-А.

Викладач:
Мельникова Н.І.

Львів – 2019 р.

Лабораторна робота № 4.

Варіант №4

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

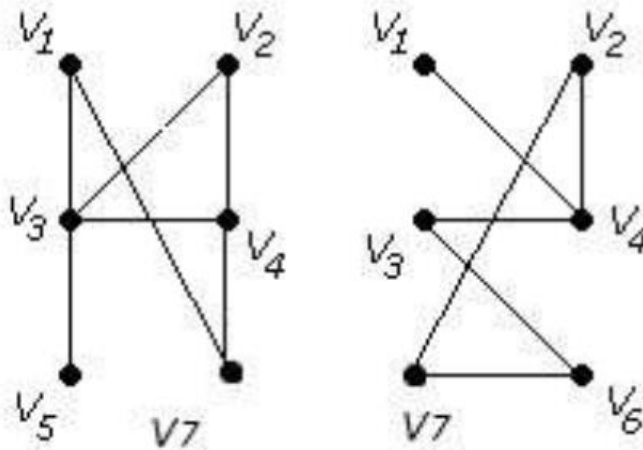
Мета роботи: набуття практичних вмінь та навичок з використанням алгоритмів Прима і Краскала.

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 G_1 та G_2 ($G_1 + G_2$),
- 4) розщепити 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.

4

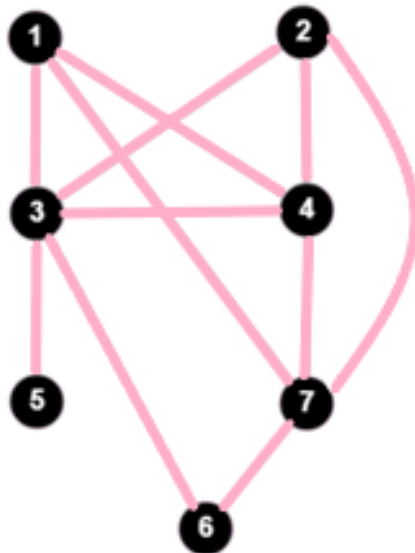


Розв'язки:

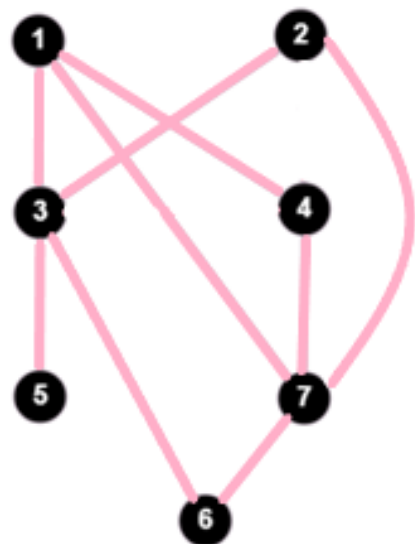
- 1) знайти доповнення до першого графу,



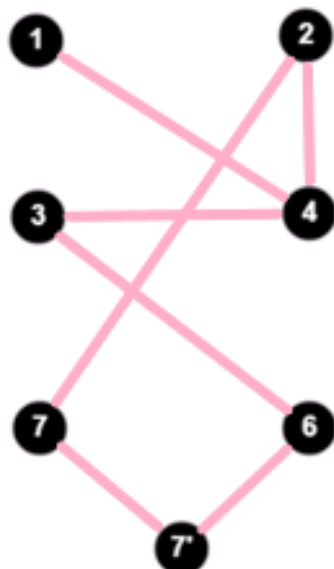
2) об'єднання графів,



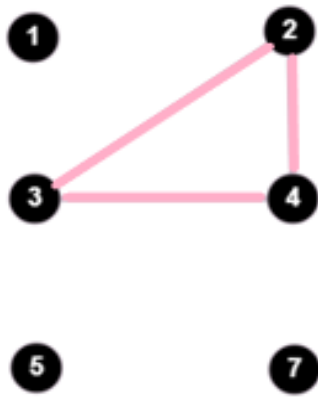
3) кільцеву суму $G1$ та $G2$ ($G1 \oplus G2$),



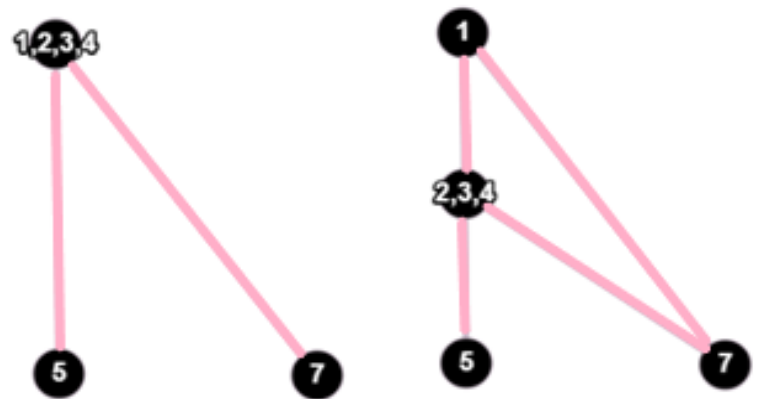
4) розщепити вершину у другому графі,



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

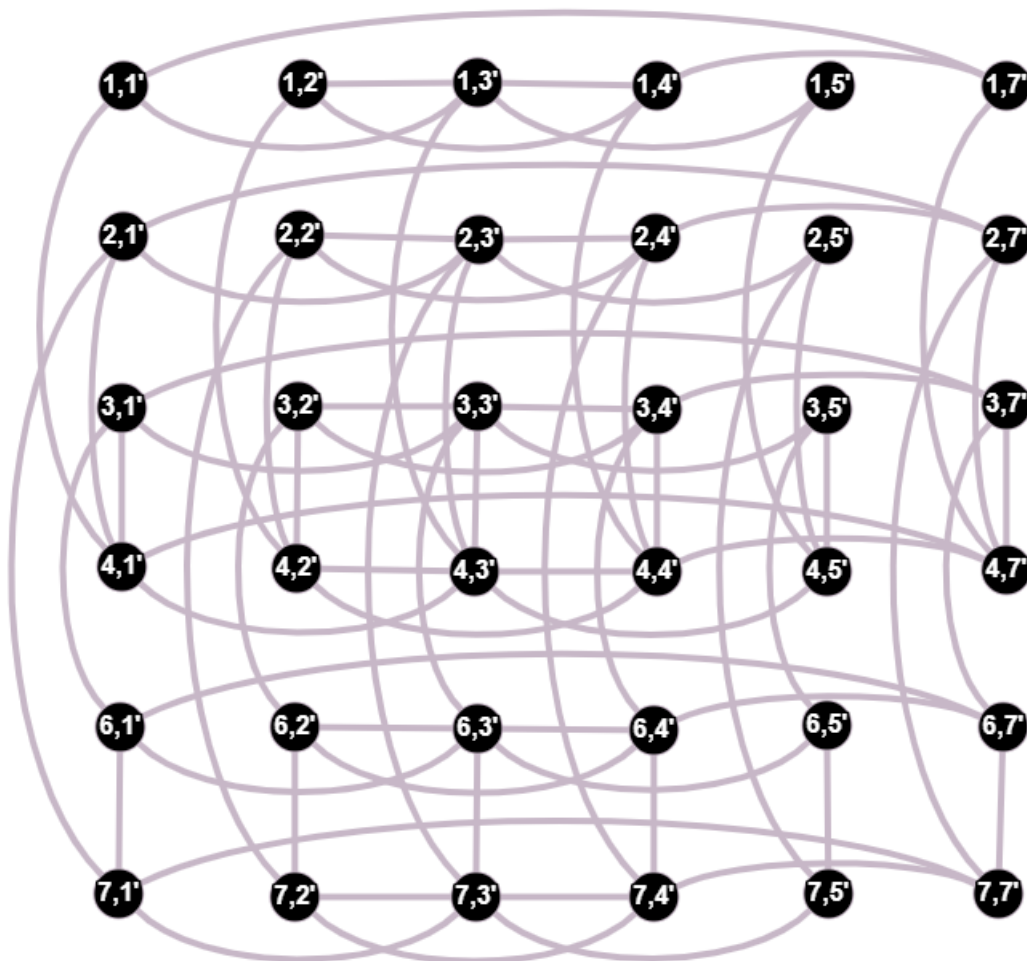


Підграф A



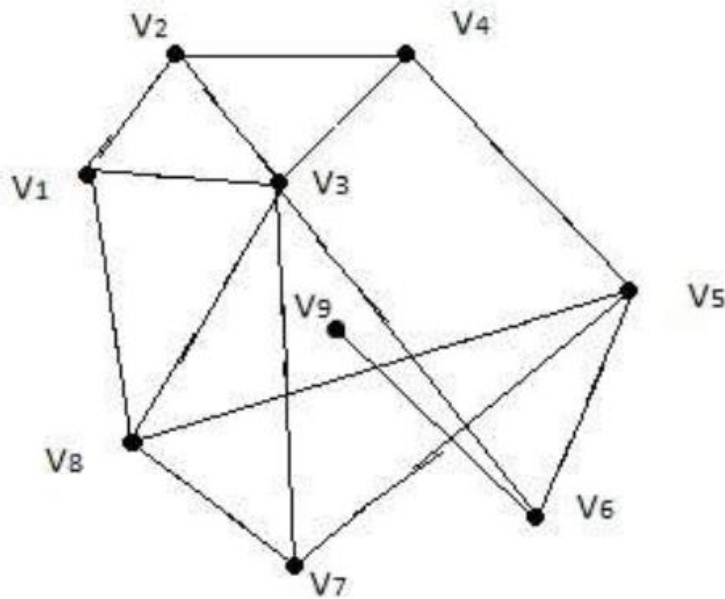
Стягнення A в G_1

6) добуток графів.



2. Знайти таблицю суміжності та діаметр графа.

4



Таблиця суміжності:

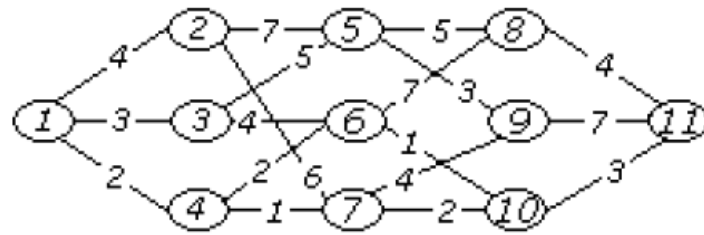
	V1	V2	V3	V4	V5	V6	V7	V8	V9
V1	0	1	1	0	0	0	0	1	0
V2	1	0	1	1	0	0	0	0	0
V3	1	1	0	1	0	1	1	1	0
V4	0	1	1	0	1	0	0	0	0
V5	0	0	0	1	0	0	1	1	0
V6	0	0	1	0	1	0	0	0	1
V7	0	0	1	0	1	0	0	1	0
V8	1	0	1	0	1	0	1	0	0
V9	0	0	0	0	0	1	0	0	0

	V1	V2	V3	V4	V5	V6	V7	V8	V9	e
V1	0	1	1	2	3	2	2	1	3	3
V2	1	0	1	1	2	2	2	2	3	3
V3	1	1	0	1	2	1	1	1	2	2
V4	2	1	1	0	1	2	2	2	3	3
V5	3	2	2	1	0	1	1	1	2	3
V6	2	2	1	2	1	0	2	2	1	2
V7	2	2	1	2	1	2	0	1	3	3
V8	1	2	1	2	1	2	1	0	3	3
V9	3	3	2	3	2	1	3	3	0	3

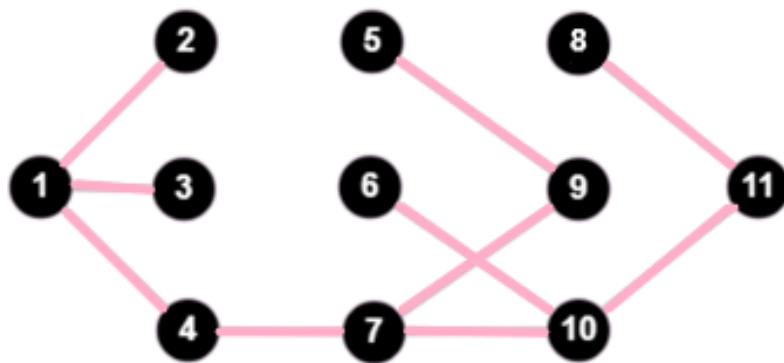
Діаметр: $D(G) = 3$

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

4



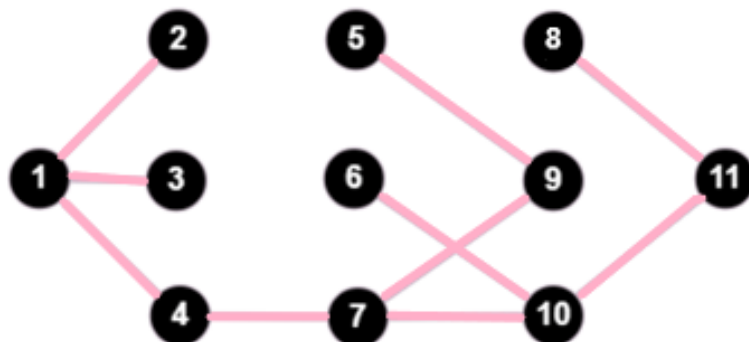
Алгоритм Прима



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

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

Алгоритм Краскала



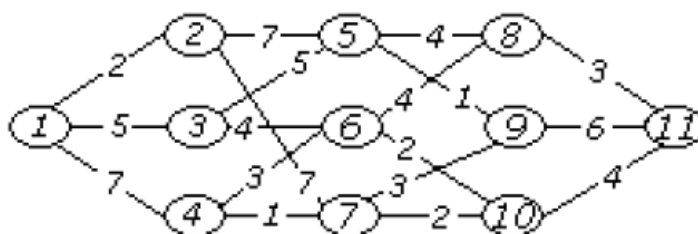
$V(t) = \{4,6,10,1,7,11,3,2,9,5,8\};$

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

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

Варіант № 4

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



```

1  #include <stdio.h>
2  #include <iostream>
3  #include <stdlib.h>
4  using namespace std;
5
6  const int q = 11;
7  int BuildTrees(int n, int A[q][q]);
8  void DeleteDuplicates(int n, int A[q][q]);
9  int InDifferTrees(int n, int A[q][q], int first, int second);
10 void AddToTheTree(int n, int A[q][q], int first, int second);
11
12 int main()
13 {
14     setlocale(LC_ALL, "Ukrainian");
15     int A[11][11] =
16     { 0, 2, 5, 7, 0, 0, 0, 0, 0, 0, 0,
17       2, 0, 0, 0, 7, 0, 7, 0, 0, 0, 0,
18       5, 0, 0, 0, 5, 4, 0, 0, 0, 0, 0,
19       7, 0, 0, 0, 0, 3, 1, 0, 0, 0, 0,
20       0, 7, 5, 0, 0, 0, 0, 4, 1, 0, 0,
21       0, 0, 4, 3, 0, 0, 0, 4, 0, 2, 0,
22       0, 7, 0, 1, 0, 0, 0, 0, 3, 2, 0,
23       0, 0, 0, 0, 4, 4, 0, 0, 0, 0, 3,
24       0, 0, 0, 0, 1, 0, 3, 0, 0, 0, 6,
25       0, 0, 0, 0, 0, 2, 2, 0, 0, 0, 4,
26       0, 0, 0, 0, 0, 0, 0, 0, 3, 6, 4, 0 };
27
28     DeleteDuplicates(11, A);
29     for (int i = 1; i <= 7; i++){
30         cout << "\nВузели з вагою: " << i << ": ";
31         for (int j = 1; j <= 11; j++){
32             for (int k = 1; k <= 11; k++){
33                 if (A[j - 1][k - 1] == i){
34                     cout << " " << j << "-" << k;;
35                 }
36             }
37         }
38     }
39     cout << "\n";

```

```

40 //Перевірте відсортовані вузли та додайте до дерева
41
42 int B[11][11];
43 BuildTrees(11, B);
44 cout << "\n\nНове дерево: "; //вага 7 - максимальна вага
45 for (int i = 1; i <= 7; i++){
46     //перший вузол
47     for (int j = 1; j <= 11; j++){
48         //другий вузол
49         for (int k = 1; k <= 11; k++){
50             if (A[j - 1][k - 1] == i && InDifferTrees(11, B, j, k)){
51                 AddToTheTree(11, B, j, k);
52                 cout << " " << j << "-" << k;
53             }
54         }
55     }
56 }
57 return 0;
58 }
59
60 void DeleteDuplicates(int n, int A[q][q]){
61     for (int i = 0; i < n; i++){
62         for (int j = 0; j < n; j++){
63             if (j < i){
64                 A[i][j] = 0;
65             }
66         }
67     }
68 }
69
70 int BuildTrees(int n, int A[q][q]) {
71     for (int i = 0; i < n; i++) {
72         for (int j = 0; j < n; j++) {
73             A[i][j] = 0;
74         }
75     }
76     for (int i = 0; i < n; i++) {
77         A[i][i] = i + 1;
78     }
79     return A[n][n];
80 }
81
82 void AddToTheTree(int n, int A[q][q], int first, int second) {
83     int scndLine;
84     for (int i = 0; i < n; i++) {
85         for (int j = 0; j < n; j++) {
86             if (A[i][j] == second) {
87                 scndLine = i;
88             }
89         }
90     }

```



```

91     for (int i = 0; i < n; i++) {
92         for (int j = 0; j < n; j++) {
93             if (A[i][j] == first) {
94                 for (int k = 0; k < n; k++) {
95                     if (A[scndLine][k]) {
96                         A[i][k] = A[scndLine][k];
97                         A[scndLine][k] = 0;
98                     }
99                 }
100             }
101         }
102     }
103 }
104
105 int InDifferTrees(int n, int A[q][q], int first, int second){
106     int temp1, temp2;
107     //Лінія
108     for (int i = 0; i < n; i++){
109         temp1 = 0;
110         temp2 = 0;
111         //перший елемент
112         for (int j = 0; j < n; j++){
113             if (A[i][j] == first){
114                 temp1 = 1;
115             }
116         }
117         //другий елемент
118         for (int k = 0; k < n; k++){
119             if (A[i][k] == second){
120                 temp2 = 1;
121             }
122         }
123         if (temp1 && temp2){
124             return 0;
125         }
126     }
127     return 1;
128 }

```

CS Консоль отладки Microsoft Visual Studio

```

Вузли з вагою: 1: 4-7 5-9
Вузли з вагою: 2: 1-2 6-10 7-10
Вузли з вагою: 3: 4-6 7-9 8-11
Вузли з вагою: 4: 3-6 5-8 6-8 10-11
Вузли з вагою: 5: 1-3 3-5
Вузли з вагою: 6: 9-11
Вузли з вагою: 7: 1-4 2-5 2-7

```

Нове дерево: 4-7 5-9 1-2 6-10 7-10 7-9 8-11 3-6 5-8 1-3

Висновок: на цій лабораторній я набув практичних вмінь та навичок з використанням алгоритмів Прима і Краскала.