

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра систем штучного інтелекту

Звіт
Лабораторна робота №4
з дисципліни:
“Дискретна математика”

Виконав:

Студент групи КН-113

Вовчак Л. В.

Викладач:

Мельникова Н.І.

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

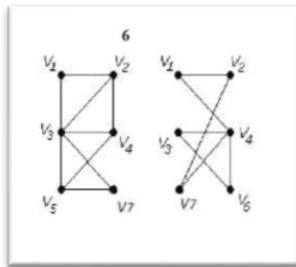
Мета: набуття практичних вмінь та навичок з використання алгоритмів Пріма і Краскала.

Постановка завдання:

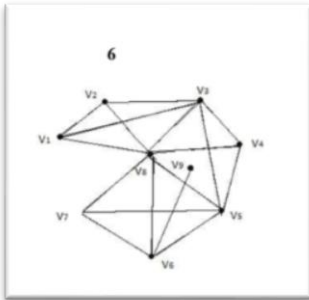
Завдання № 1.

Розв'язати на графах наступні задачі:

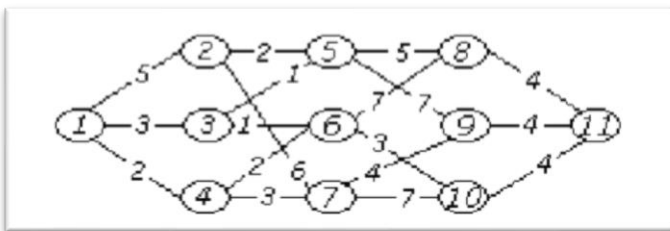
1. Виконати наступні операції над графами:
 - 1) знайти доповнення до першого графу,
 - 2) об'єднання графів,
 - 3) кільцеву суму $G1$ та $G2$ ($G1+G2$),
 - 4) розщепити вершину у другому графі,
 - 5) виділити підграф A , що складається з 3-х вершин в $G1$ і знайти стягнення A в $G1$ ($G1 \setminus A$),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



2. Знайти таблицю суміжності та діаметр графа.



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

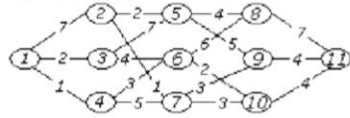


Завдання №2.

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

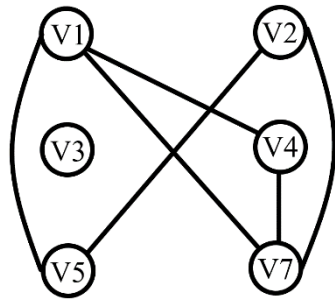
Варіант № 6

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

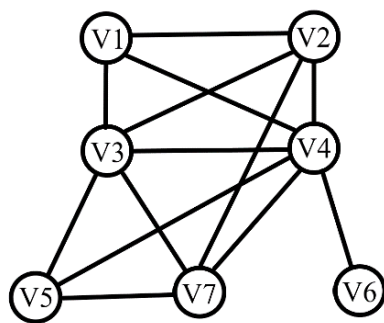


Розв'язок задачі №1

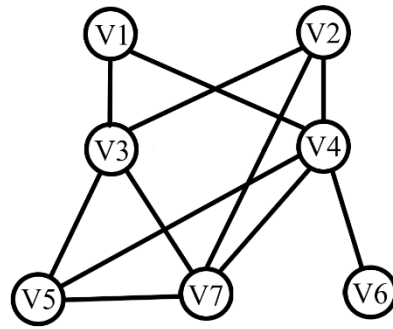
11)Доповнення до G1



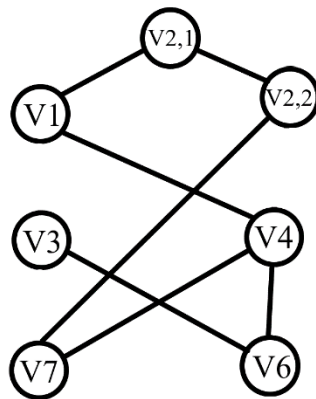
2)



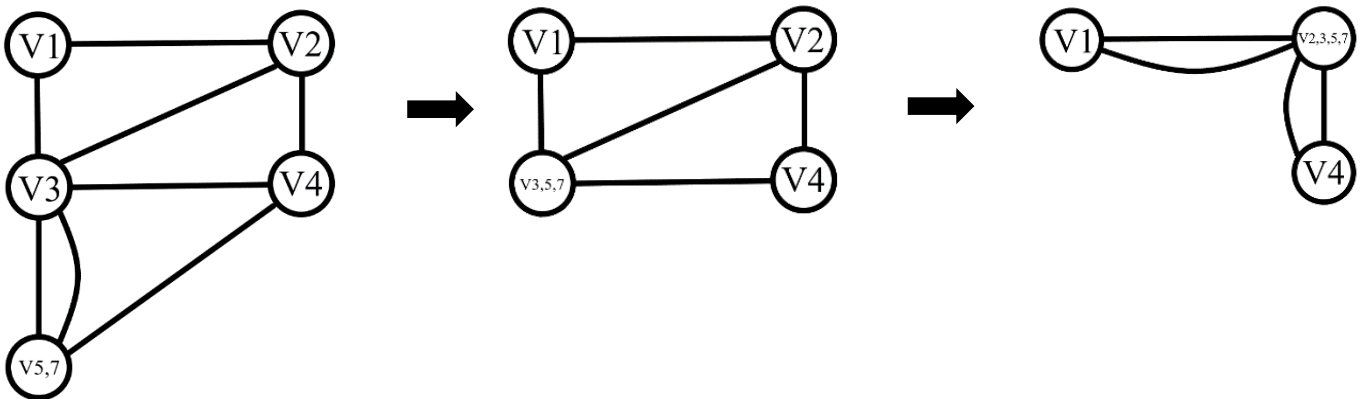
3)



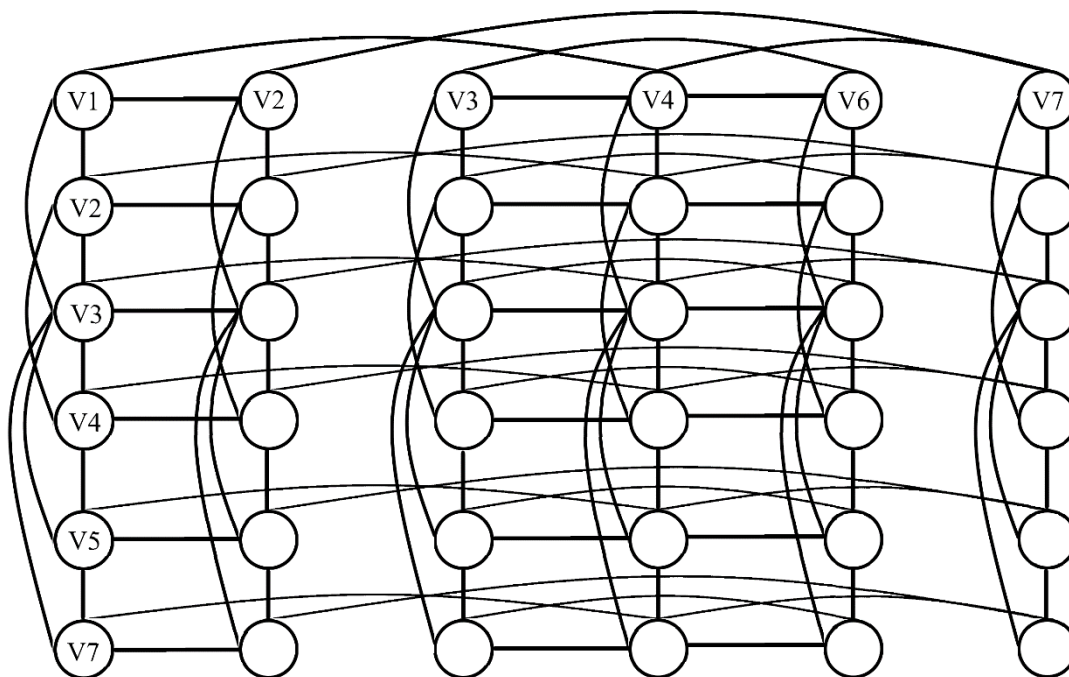
4)



5) $A = \{V3, V5, V7\}$



6)



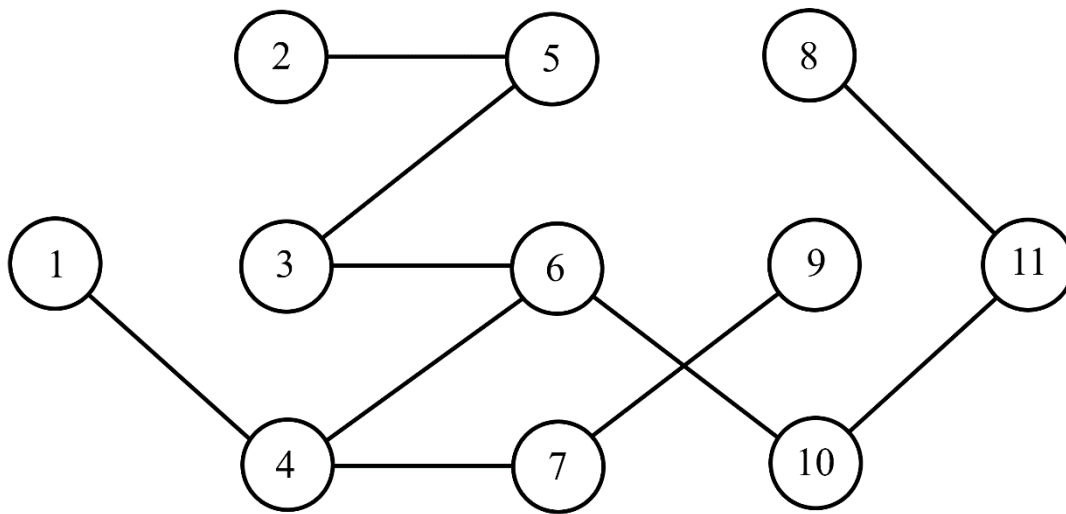
2.

1) 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	0	0	0	0	1	0
V3	1	1	0	1	1	0	0	1	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

2) Діаметр графа = 3;

3.1 Метод Краскала

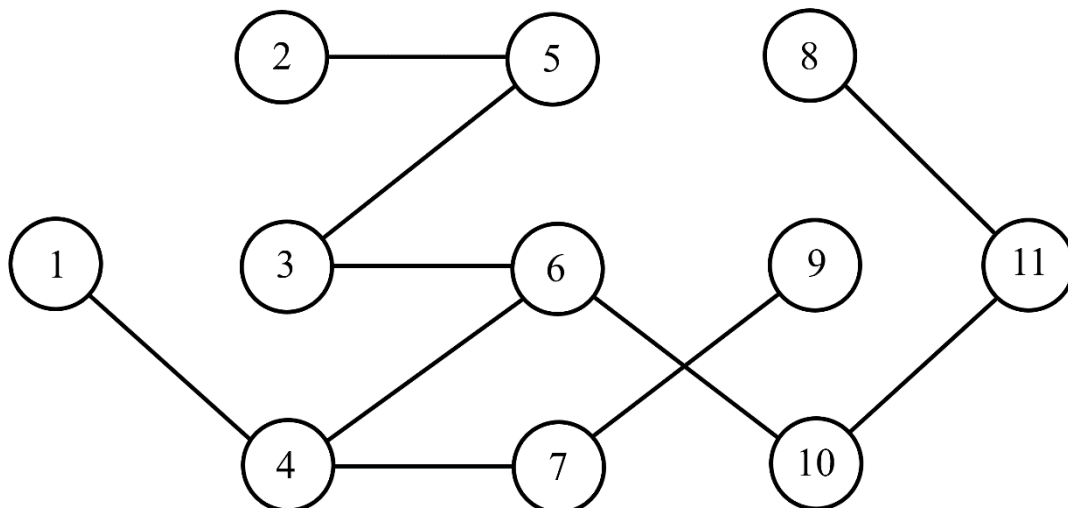


$V = \{3, 5, 6, 2, 1, 4, 7, 10, 9, 11, 8\};$

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

Загальна довжина = 26;

3.2 Метод Прима



Загальна довжина = 26;

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

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

Розв'язок задачі №2

Код програмної реалізації

```
#include <cstdio>
#include <iostream>

#define size 11

using namespace std;

void Del_duplicats(int Arr[size][size]) {
    for (int i = 0; i < size; i++) {
        for (int j = 0; j < size; j++) {
            if (j < i) {
                Arr[i][j] = 0;
            }
        }
    }
}

int Create(int A[size][size]){
    for (int i = 0; i < size; i++)
    {
        for (int j = 0; j < size; j++)
        {
            A[i][j] = 0;
        }
    }
    for (int i = 0; i < size; i++)
    {
        A[i][i] = i + 1;
    }
    return A[size][size];
}

int NotOne(int n, int A[size][size],int f, int s) {
    int t1,t2;
    for (int i = 0; i < size; i++) {
```

```

    t1 = t2 = 0;
    for (int j = 0; j < size; j++) {
        if (A[i][j] == f) {
            t1 = 1;
        }
    }

    for (int k = 0; k < size; k++) {
        if (A[i][k] == s) {
            t2 = 1;
        }
    }

    if (t1 && t2) {
        return 0;
    }
    return 1;
}

void Add(int n, int Arr[size][size], int f, int s) {
    int scndLine;

    for (int i = 0; i < size; i++) {
        for (int j = 0; j < size; j++) {
            if (Arr[i][j] == s) {
                scndLine = i;
            }
        }
    }

    for (int i = 0; i < size; i++) {
        for (int j = 0; j < size; j++) {
            if (Arr[i][j] == f) {
                for (int p = 0; p < size; p++) {
                    if (Arr[scndLine][p]) {
                        Arr[i][p] = Arr[scndLine][p];
                        Arr[scndLine][p] = 0;
                    }
                }
            }
        }
    }
}

```



```

    }
  }
}
}

```

```

int main()
{
  int A[size][size] = {
    { 0, 2, 5, 7, 0, 0, 0, 0, 0, 0, 0 },
    { 2, 0, 0, 0, 7, 0, 7, 0, 0, 0, 0 },
    { 5, 0, 0, 0, 5, 4, 0, 0, 0, 0, 0 },
    { 7, 0, 0, 0, 0, 3, 1, 0, 0, 0, 0 },
    { 0, 7, 5, 0, 0, 0, 0, 4, 1, 0, 0 },
    { 0, 0, 4, 3, 0, 0, 0, 4, 0, 2, 0 },
    { 0, 7, 0, 1, 0, 0, 0, 0, 3, 2, 0 },
    { 0, 0, 0, 0, 4, 4, 0, 0, 0, 0, 3 },
    { 0, 0, 0, 0, 1, 0, 3, 0, 0, 0, 6 },
    { 0, 0, 0, 0, 0, 2, 2, 0, 0, 0, 4 },
    { 0, 0, 0, 0, 0, 0, 0, 3, 6, 4, 0 }
  };

```

```

Del_duplicats(A);

```

```

for (int k = 1; k <= 7; k++)
{
  cout << endl << "Edges with weight " << k << ": ";
  for (int i = 1; i <= size; i++){
    for (int j = 1; j <= size; j++){
      if (A[i - 1][j - 1] == k)
      {
        cout << "(" << i << "; " << j << ") ";
      }
    }
  }
}
}

```

```

int B[size][size];
Create(B);

```

```

cout << endl << endl << "Tree:";

for (int k = 1; k <= 7; k++)
{
    for (int i = 1; i <= size; i++) {
        for (int j = 1; j <= size; j++) {
            if (A[i - 1][j - 1] == k && NotOne(size,B,i,j))
            {
                Add(size,B,i,j);
                cout << "(" << i << ";" << j << ")" ";
            }
        }
    }
}
cout << endl;
return 0;
}

```

Результат програми

```

Edges with weight 1: (4;7) (5;9)
Edges with weight 2: (1;2) (6;10) (7;10)
Edges with weight 3: (4;6) (7;9) (8;11)
Edges with weight 4: (3;6) (5;8) (6;8) (10;11)
Edges with weight 5: (1;3) (3;5)
Edges with weight 6: (9;11)
Edges with weight 7: (1;4) (2;5) (2;7)

Tree:(4;7) (5;9) (1;2) (6;10) (7;10) (7;9) (8;11) (3;6) (5;8) (1;3)

Process returned 0 (0x0)   execution time : 0.042 s
Press any key to continue.

```

Висновок: Виконуючи дану лабораторну роботу, я отримав навички в роботі з основними операціями над графами та знаходженні остова мінімальної ваги за алгоритмом Пріма-Краскала.