

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА
ПОЛІТЕХНІКА”**

Кафедра систем штучного інтелекту

Лабораторна робота № 4

з дисципліни

«Дискретна математика»

Виконав:

Студент групи КН-113

Волошин Володимир

Викладач:

Мельникова Н.І.

Львів – 2019р.

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

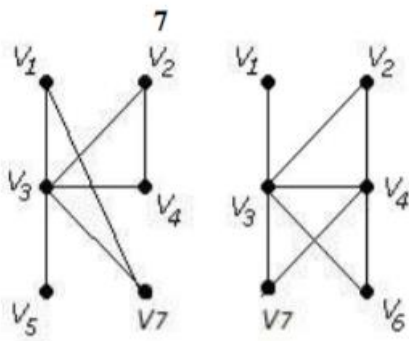
Мета роботи: Набуття практичних вмінь та навичок з використання алгоритмів Пріма і Краскала.

Варіант № 7

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

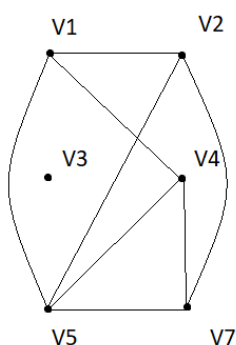
1. Виконати наступні операції над графами:

- 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.

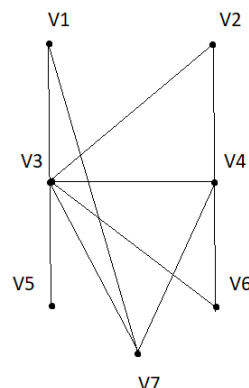


Розв'язок:

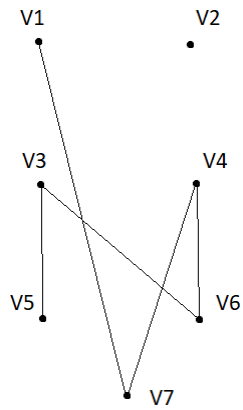
1)



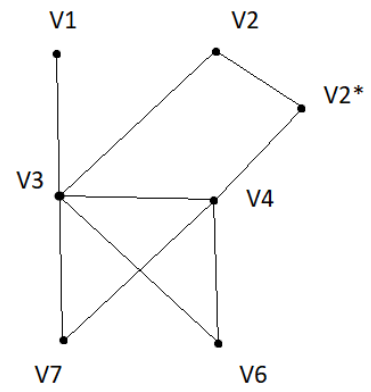
2)



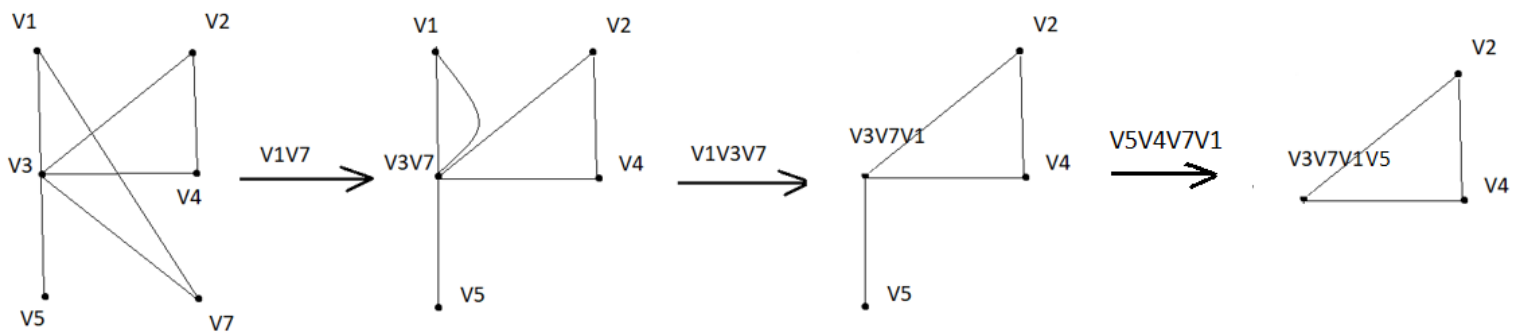
3)



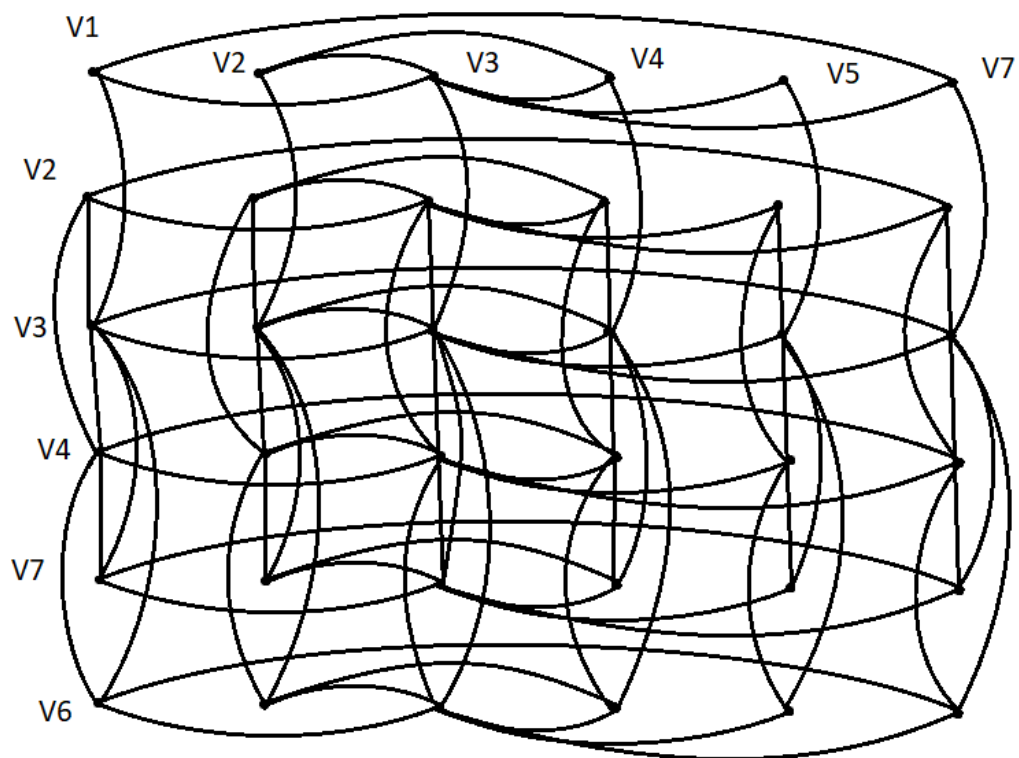
4)



5) Виберемо підграф А з вершинами V1, V7, V5. Тоді $G1 \setminus A$ буде:

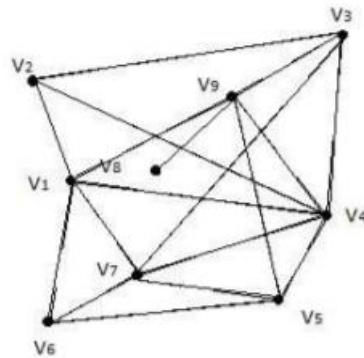


6)



2. Знайти таблицю суміжності та діаметр графа.

7



Розв'язок:

Таблиця суміжності:

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

Діаметр = 3.

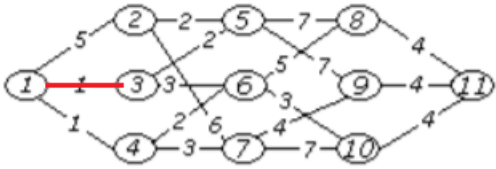
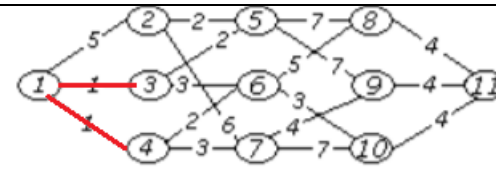
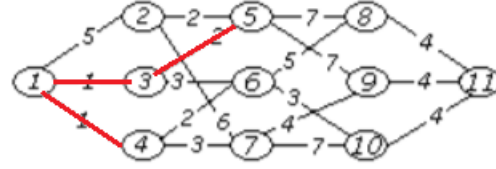
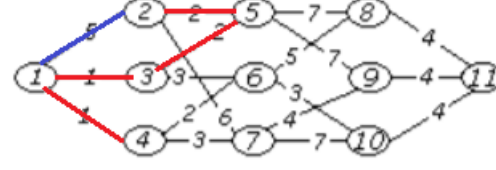
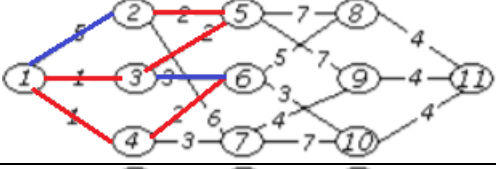


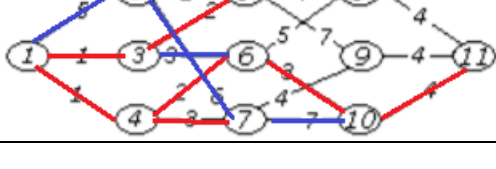
3. Знайти двома методами (Краскала і Прима) мінімальне остове дерево графа.

7



Розв'язок:

Метод Краскала

Крок	Малюнок	Пояснення	Вибрані ребра
1		Вибирається ребро з найменшою вагою. Серед ребер (1,3) та (1,4) довільним чином було вибрано перше.	(1,3)
2		Тепер найменшим є (1,4). Його і вибираємо. Таким же чином будемо діяти і далі.	(1,3), (1,4)
3		Було вибрано ребро (3,5)	(1,3), (1,4), (3,5)
4		Було вибрано ребро (2,5), а ребро яке утворить цикл, тобто (1,2), одразу закреслюємо синім кольором і воно надалі розглядається не буде. Будемо так діяти і далі.	(1,3), (1,4), (3,5), (2,5)
5		Було вибрано ребро (4,6), а (3,6) закреслено	(1,3), (1,4), (3,5), (2,5), (4,6)
6		Обираємо (4,7)	(1,3), (1,4), (3,5), (2,5), (4,6), (4,7)
7		Було вибрано ребро (6,10), а (7,10) та (2,7) закреслено	(1,3), (1,4), (3,5), (2,5), (4,6), (4,7), (6,10)
8		Було вибрано ребро (10,11)	(1,3), (1,4), (3,5), (2,5), (4,6), (4,7), (6,10), (10,11)

9		Було вибрано ребро (9,11), а (7,9) і (5,9) закреслено	(1,3),(1,4),(3,5),(2,5), (4,6),(4,7),(6,10),(10,11), (9,11)
10		Було вибрано ребро (8,11), а (8,6) і (5,7) закреслено.	(1,3),(1,4),(3,5),(2,5), (4,6),(4,7),(6,10),(10,11), (9,11), (8,11)

Метод Прима

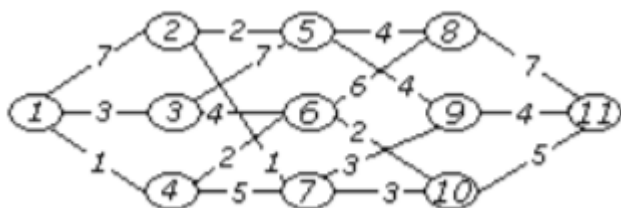
Крок	Малюнок	Пояснення	Вибрані ребра
1		Вибираємо довільну точку та прилегле ребро з найменшою вагою. Нехай це буде точка 1 і ребро (1,3)	(1,3)
2		Тепер серед іцидентних ребер обираємо ребро з найменшою вагою. Так будемо діяти і надалі. Обрано ребро(1,4)	(1,3),(1,4)
3		Обрано ребро (3,5)	(1,3),(1,4),(3,5)
4		Обрано ребро (2,5), а ребро (1,2) закреслено, оскільки воно утворить цикл. Так будемо діяти і надалі.	(1,3),(1,4),(3,5),(2,5)
5		Було вибрано ребро (4,6), а ребро (3,6) закреслено .	(1,3),(1,4),(3,5),(2,5), (4,6)
6		Було вибрано ребро (6,10).	(1,3),(1,4),(3,5),(2,5), (4,6),(6,10)

7		Було вибрано ребро (4,7), а ребра (2,7) та (7,10) закреслено.	(1,3),(1,4),(3,5),(2,5), (4,6),(6,10),(4,7)
8		Було вибрано ребро (7,9).	(1,3),(1,4),(3,5),(2,5), (4,6),(6,10),(4,7), (7,9),
9		Було вибрано ребро (10,11), а ребро (9,11) закреслено.	(1,3),(1,4),(3,5),(2,5), (4,6),(6,10),(4,7), (7,9),(10,11)
10		Було вибрано ребро (8,12), а ребро (6,8) та (5,8) закреслено.	(1,3),(1,4),(3,5),(2,5), (4,6),(6,10),(4,7), (7,9),(10,11),(8,11)

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

Варіант № 7

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



Програмна реалізація:

```
#include <iostream>
#include <fstream>
using namespace std;

//Функція виводу масиву на екран
void PrintArray(int** arr, int size)
{
    for (int i = 0; i < size; i++)
    {
        for (int j = 0; j < size; j++)
        {
            if (arr[i][j] != 0)
                cout << "|" << arr[i][j] << "\\t";
            else
                cout << arr[i][j] << "\\t";
        }
        cout << endl<<endl;
    }
    cout <<
    "=====\\n";
}

//Функція знаходження мінімального значення
int FindMin(int* arr, int size)
{
    for (int i = 1; i < size; i++){
        if (arr[0] > arr[i])
            arr[0] = arr[i];
    }
    return arr[0];
}

int main()
{
    setlocale(LC_ALL, "Ukr");
    int size = 11;
    int** arr1 = new int* [size]; //створення першого двовимірного динамічного масиву
    for (int i = 0; i < size; i++)
        arr1[i] = new int[size];

    int** arr2 = new int* [size]; //створення другого двовимірного динамічного масиву
    for (int i = 0; i < size; i++)
        arr2[i] = new int[size];

    for (int i = 0; i < size; i++){ //Ініціалізація масивів нулями
        for (int j = 0; j < size; j++){
            arr1[i][j] = 0;
            arr2[i][j] = 0;
        }
    }

    int num_of_edges = 18;
    int weight = 0, vertex1 = 0, vertex2 = 0;

    ifstream fin("FILE.txt"); //Зчитування даних про граф з файлу
    if (!fin.is_open())
        cout << "Error\\n";
    else {
        for (int i = 0; i < num_of_edges; i++){
            fin >> weight;
            fin >> vertex1;
            fin >> vertex2;
            arr1[vertex1 - 1][vertex2 - 1] = weight;
            arr1[vertex2 - 1][vertex1 - 1] = weight;
        }
    }
}
```



```

}
fin.close();
/*for (int i = 0; i < num_of_edges; i++) // Ввід даних вручну
{
    cin >> weight;
    cin >> vertex1;
    cin >> vertex2;
    arr1[vertex1 - 1][vertex2 - 1] = weight;
    arr1[vertex2 - 1][vertex1 - 1] = weight;
}*/
PrintArray(arr1, size);

int min[100], min_size = 0, counter = 0, step = 1, n_of_chosens = 1;
int* chosens = new int [size]; //Створення динамічного масиву
chosens[counter] = 0;
bool end = false;
while(!end)
{
    end = true;
    min_size = 0;
    for (int n = 0; n < size; n++) //Видалення потрібних рядків рядків
        arr1[chosens[counter]][n] = 0;

    for (int j = 0; j < n_of_chosens; j++) {
        for (int i = 0; i < size; i++) { //Пошук можливих наступних ребер
            if (arr1[i][chosens[j]] > 0) {
                min[min_size] = arr1[i][chosens[j]];
                min_size++;
                end = false;
            }
        }
    }
    int minimum = FindMin(min, min_size); //Знаходження мінімального значення з них
    for (int j = 0; j < n_of_chosens; j++) {
        for (int i = 0; i < size; i++) {
            if (arr1[i][chosens[j]] == minimum) { //Прирівнювання значень та
знаходження вибраного ребра та номеру рядка
                arr2[i][chosens[j]] = arr1[i][chosens[j]];
                arr2[chosens[j]][i] = arr1[i][chosens[j]];
                n_of_chosens++;
                counter++;
                chosens[counter] = i;
                goto link;
            }
        }
    }
link:
    cout << "Step " << step << endl;
    step++;
    PrintArray(arr2, size);
}
cout << "\nМінімальне остове дерево:";
for (int i = 0; i < size; i++) { //Виведення мінімального остового дерева
    for (int j = i; j < size; j++) {
        if (arr2[i][j] > 0)
            cout << "(" << i + 1 << ", " << j + 1 << ") ";
    }
}
return 0;
}

```

Результат виконання програми :

```
=====
Step 11
0      0      |3|      |1|      0      0      0      0      0      0      0
0      0      0      0      |2|      0      |1|      0      0      0      0
|3|     0      0      0      0      0      0      0      0      0      0
|1|     0      0      0      0      |2|      0      0      0      0      0
0      |2|     0      0      0      0      0      |4|      0      0      0
0      0      0      |2|      0      0      0      0      0      |2|      0
0      |1|     0      0      0      0      0      0      |3|      |3|      0
0      0      0      0      |4|      0      0      0      0      0      0
0      0      0      0      0      0      |3|      0      0      0      |4|
0      0      0      0      0      |2|      |3|      0      0      0      0
0      0      0      0      0      0      0      0      |4|      0      0
=====
Мінімальне остове дерево:(1,3) (1,4) (2,5) (2,7) (4,6) (5,8) (6,10) (7,9) (7,10) (9,11)
```