**Лабораторна робота №4**

**Тема:**ОСНОВНІ ОПЕРАЦІЇ НАД ГРАФАМИ. ЗНАХОДЖЕННЯ ОСТОВА МІНІМАЛЬНОЇ ВАГИ ЗА АЛГОРИТМОМ ПРІМА-КРАСКАЛА.

*Мета:*навчитися виконувати операції над графами, навчитися знаходити острова мінімальної ваги за алгоритмом Прима-Краскала.

**Зміст роботи:**

**Завдання №1** Використовуючи теоретичні відомості, розв’язати на графах наступні задачі за своїм варіантом:

***Завдання 1.1:***

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 А, що складається з 3-х вершин в G1 і знайти

стягнення А в G1 (G1\ A),

6) добуток графів.

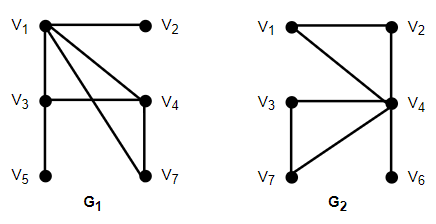


Рис 1.1 Зображення двох графів, над якими треба виконати операції.

1. Знайти доповнення до першого графу G1.

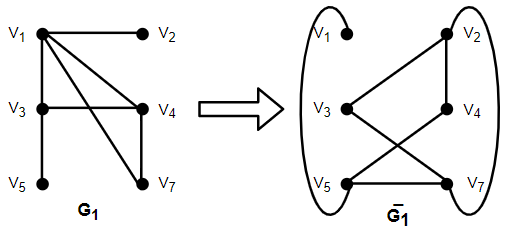


Рис 1.2 Зображення доповнення до першого графа.

1. Знайти об’єднання графів G1 та G2.

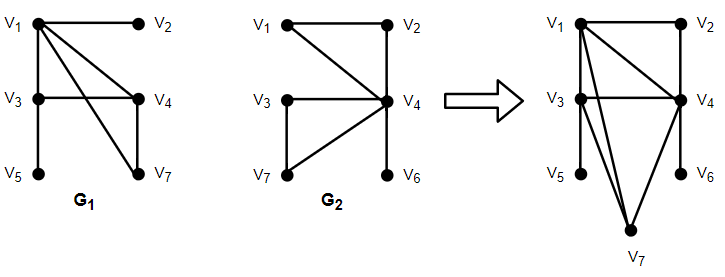


Рис 1.3 Зображення об`єднання графів G1 та G2.

1. Знайти кільцеву суму G1 та G2 (G1+G2).

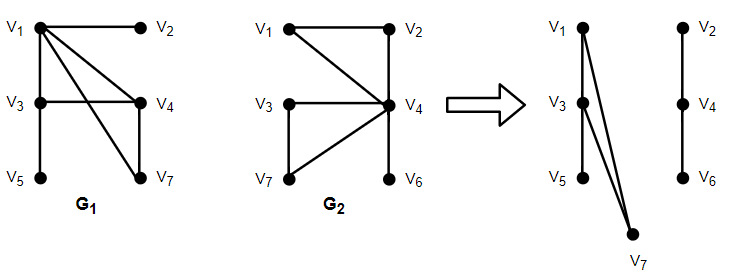


Рис 1.4 Зображення кільцевої суми двох графів.

G1 = (V1, E1): V1={V1, V2, V3, V4,V5,V7}, E1={(V1,V2), (V1,V3), (V1,V4), (V1,V7), (V3,V4), (V3,V5), (V4,V7)}

G2 = (V2, E2): V2={V1, V2, V3, V4,V6,V7}, E2={(V1,V2), (V1,V4), (V2,V4), (V3,V4), (V3,V7), (V4,V6), (V4,V7)}

G = (V, E) = , у якому V = V1 ∪ V2 ; E = E1 ∆ E2 = ( E1 ∪ E2 ) \ ( E1 ∩ E2 )

V={V1, V2, V3, V4,V5,V6,V7}, E={(V1,V3), (V1,V7), (V2,V4),(V3,V5), (V3,V7), (V4,V6)}

1. Знайти розщепити вершину у другому графі G2.

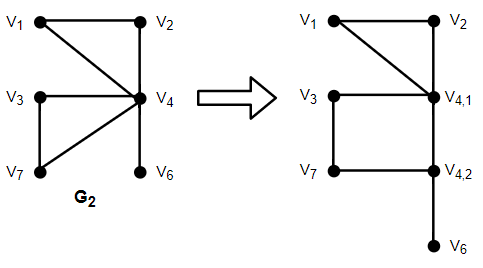


Рис 1.5 Зображення розщеплення вершини V4.

1. Виділити підграф А, що складається з 3-х вершин в G1 і знайти

стягнення А в G1 (G1\ A).

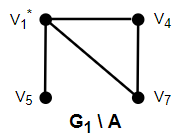
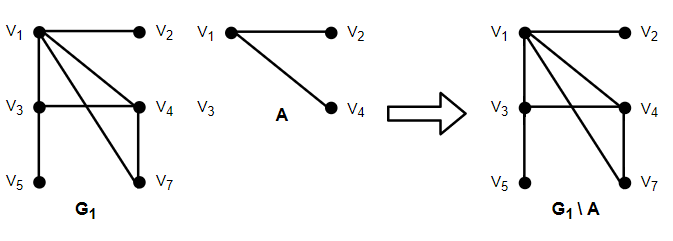
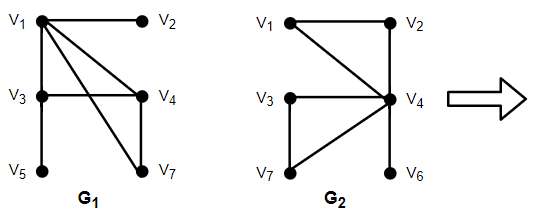


Рис 1.6 Зображення стягнення підграфа A.

1. Знайти добуток графів.



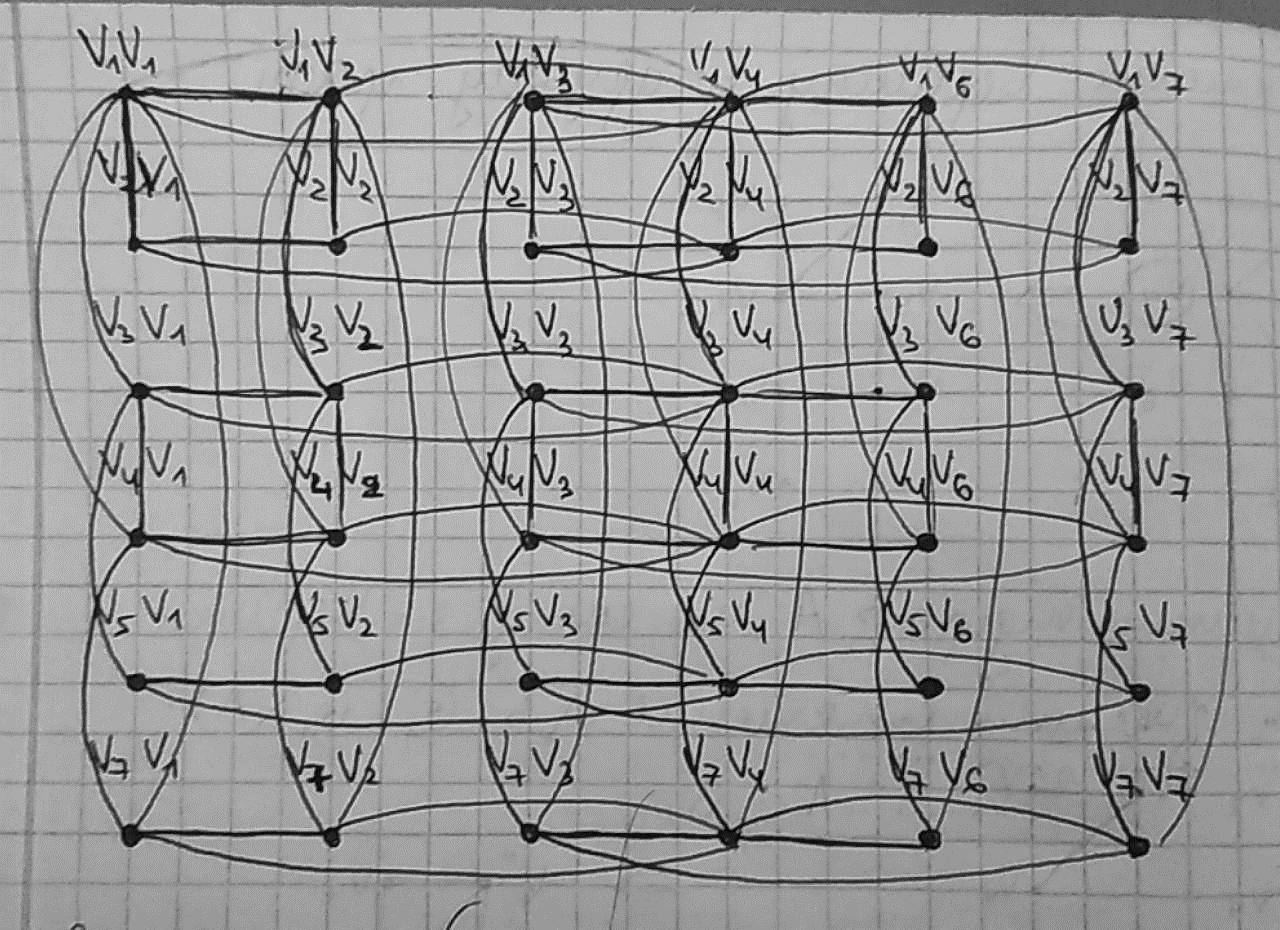


Рис 1.7 Зображення добутку двох графів.

***Завдання 1.2:*** Знайти таблицю суміжності та діаметр графа.

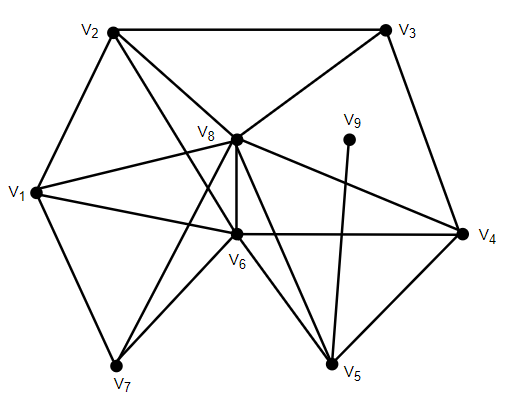


Рис 1.8 Зображення графа для якого потрібно скласти таблицю суміжності (табл. 2.1) та знайти діаметр.

*Таблиця 1.1*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | v1 | v2 | v3 | v4 | v5 | v6 | v7 | v8 | v9 |
| v1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| v2 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| v3 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| v4 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| v5 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| v6 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| v7 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| v8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| v9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

*Таблиця 1.2*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | v1 | v2 | v3 | v4 | v5 | v6 | v7 | v8 | v9 |
| v1 | 0 | 1 | 2 | 2 | 2 | 1 | 1 | 1 | 3 |
| v2 | 1 | 0 | 1 | 2 | 2 | 1 | 2 | 1 | 3 |
| v3 | 2 | 1 | 0 | 1 | 2 | 2 | 2 | 1 | 3 |
| v4 | 2 | 2 | 1 | 0 | 1 | 1 | 2 | 1 | 2 |
| v5 | 2 | 2 | 2 | 1 | 0 | 1 | 2 | 1 | 1 |
| v6 | 1 | 1 | 2 | 1 | 1 | 0 | 1 | 1 | 2 |
| v7 | 1 | 2 | 2 | 2 | 2 | 1 | 0 | 1 | 3 |
| v8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 2 |
| v9 | 3 | 3 | 3 | 2 | 1 | 2 | 3 | 2 | 0 |

Діаметр: D(G) = 3

***Завдання 1.3:*** Знайти двома методами (Краскала і Прима) мінімальне остове дерево графа.

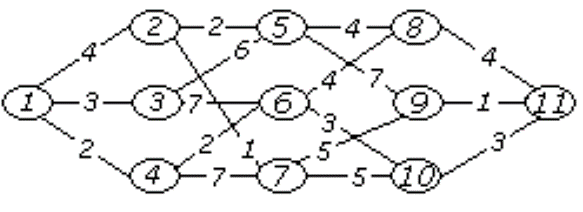
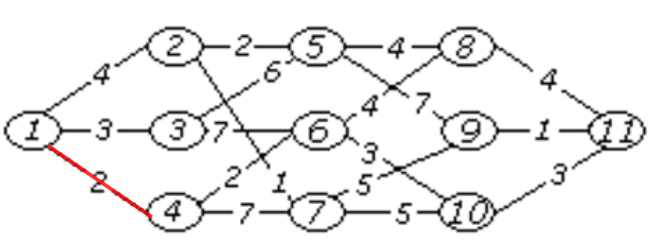
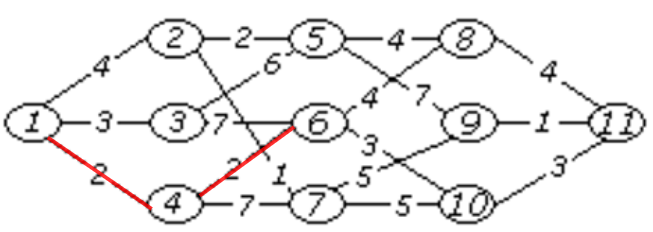
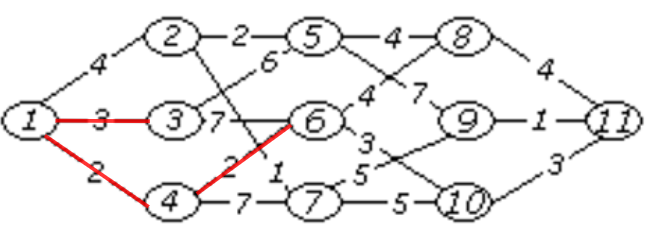


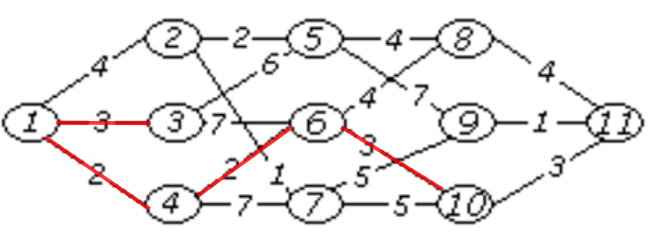
Рис 1.9 Зображення графа для якого потрібно знайти мінімальне остове дерево.

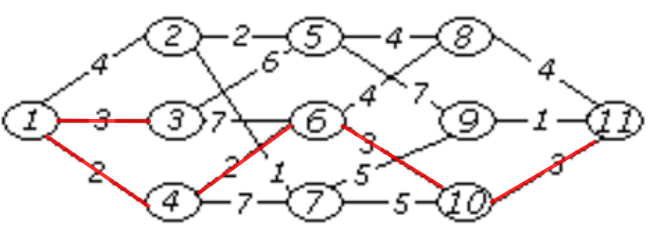
1. Метод Прима:

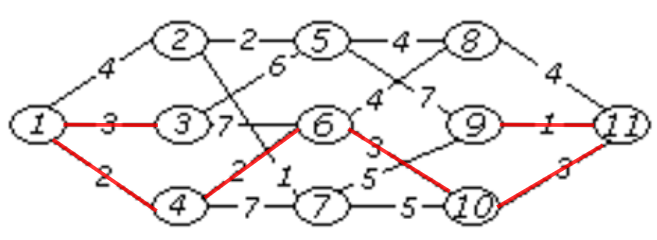


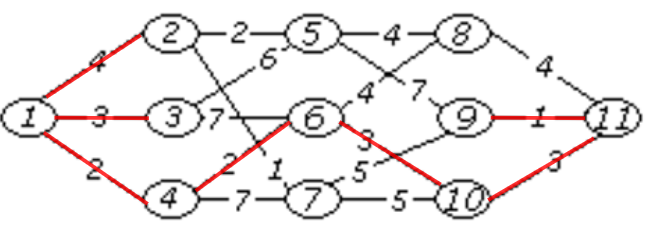


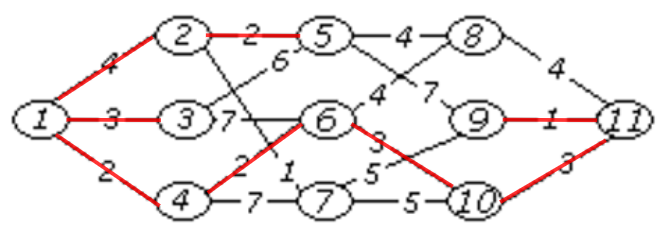


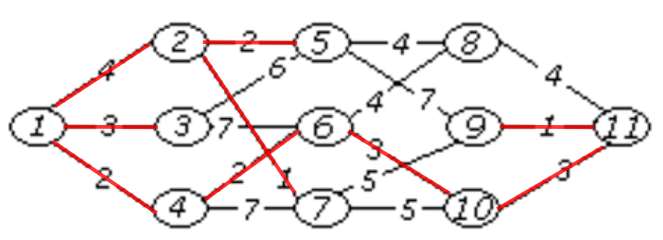


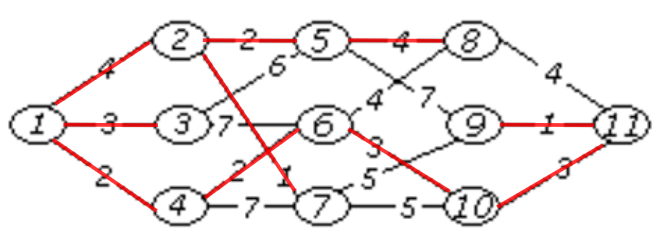






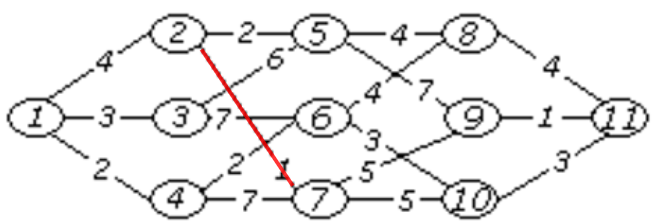


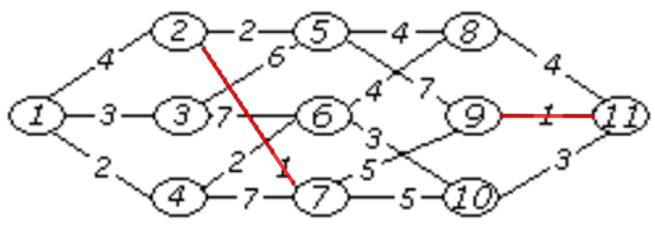


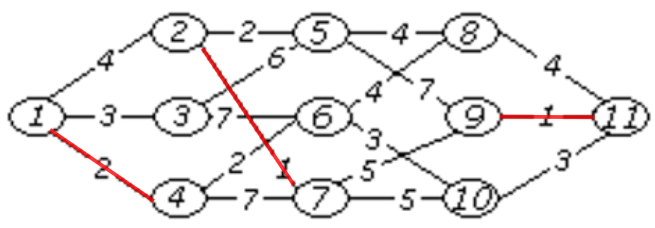


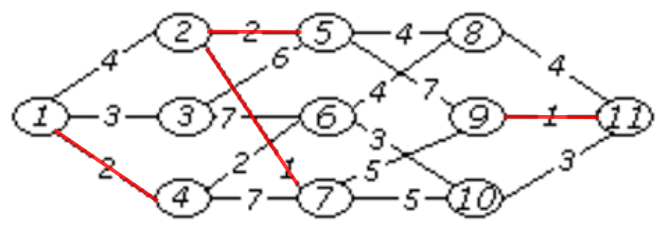
2 + 2 + 3 + 3 + 3 + 1 + 4 + 2 + 1 + 4 = 25

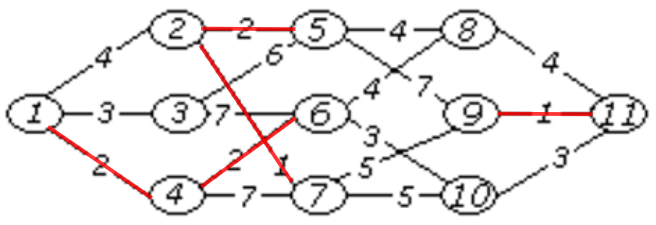
1. Метод Краскала:

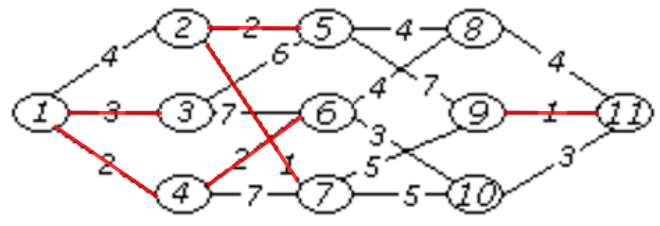


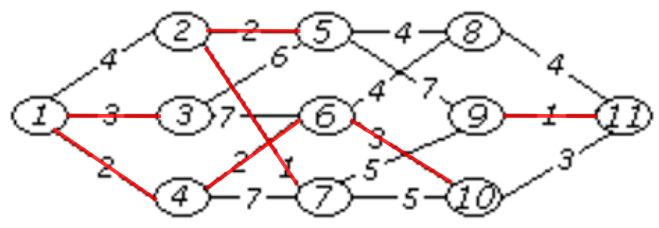


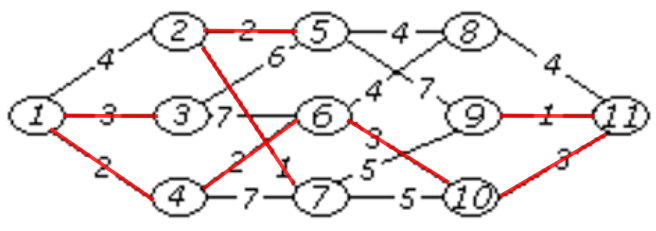


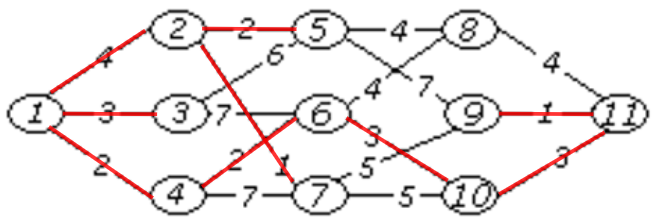


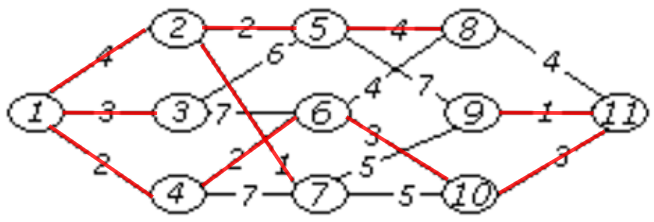












1 + 1 + 2 + 2 + 2 + 3 + 3 + 3 + 4 + 4 = 25

**Завдання №2**

Написати програму, яка реалізує алгоритм знаходження остового дерева мінімальної ваги за алгоритмом Прима чи Краскала. Етапи розв'язання задачі виводити на екран. Протестувати розроблену програму на задачі 3 із завдання № 1:

Лістинг фрагмента коду програми (метод Краскала):

private void findMinSpanTreeBtn\_Click(object sender, RoutedEventArgs e)

{

statusLabel.Content = "Підрахунок...";

LaunchMinSpanTreeTask();

}

// алгоритм Краскала

private void LaunchMinSpanTreeTask()

{

Task.Factory.StartNew(() =>

FindMinSpanTree()

)

.ContinueWith((task) =>

{

foreach (Edge e in \_mst)

{

e.Visited = true;

PaintEdge(e);

PaintNode(e.FirstNode);

PaintNode(e.SecondNode);

}

statusLabel.Content = "Повна вартість: " + task.Result.ToString("n0");

},

TaskScheduler.FromCurrentSynchronizationContext());

}

private double FindMinSpanTree()

{

\_mst.Clear();

List<Node> forest = new List<Node>();

\_edges.Sort();

double totalCost = 0;

foreach (Edge currentEdge in \_edges)

{

if (\_clusters.Count == 1)

break;

Cluster cluster1 = currentEdge.FirstNode.Cluster;

Cluster cluster2 = currentEdge.SecondNode.Cluster;

if (cluster1.Label != cluster2.Label)

{

\_mst.Add(currentEdge);

totalCost += currentEdge.Length;

currentEdge.FirstNode.Visited = true;

currentEdge.SecondNode.Visited = true;

MergeClusters(cluster1, cluster2);

}

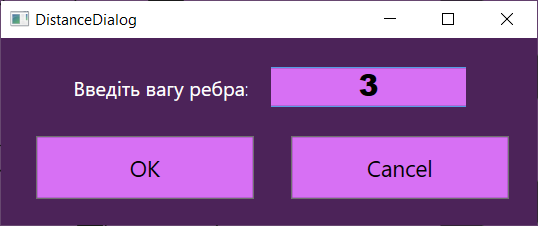
}

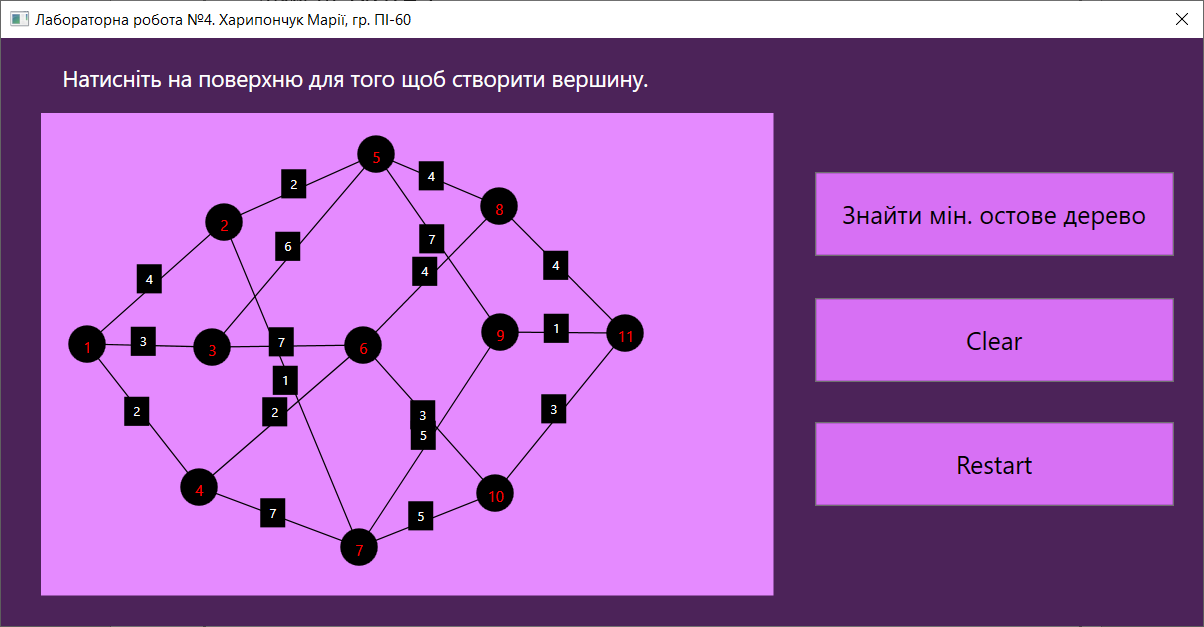
return totalCost;

}

Встановлюємо вершини лівою кнопкою миші відповідно малюнку, та задаємо відстань затиснувши клавішу CTRL та обравши дві вершини, в результаті маємо: Застосуємо алгоритм Краскала для знаходження остового дерева мінімальної ваги. Натиснувши відповідну клавішу для обрахунку отримуємо:

Демонстрація роботи програми:





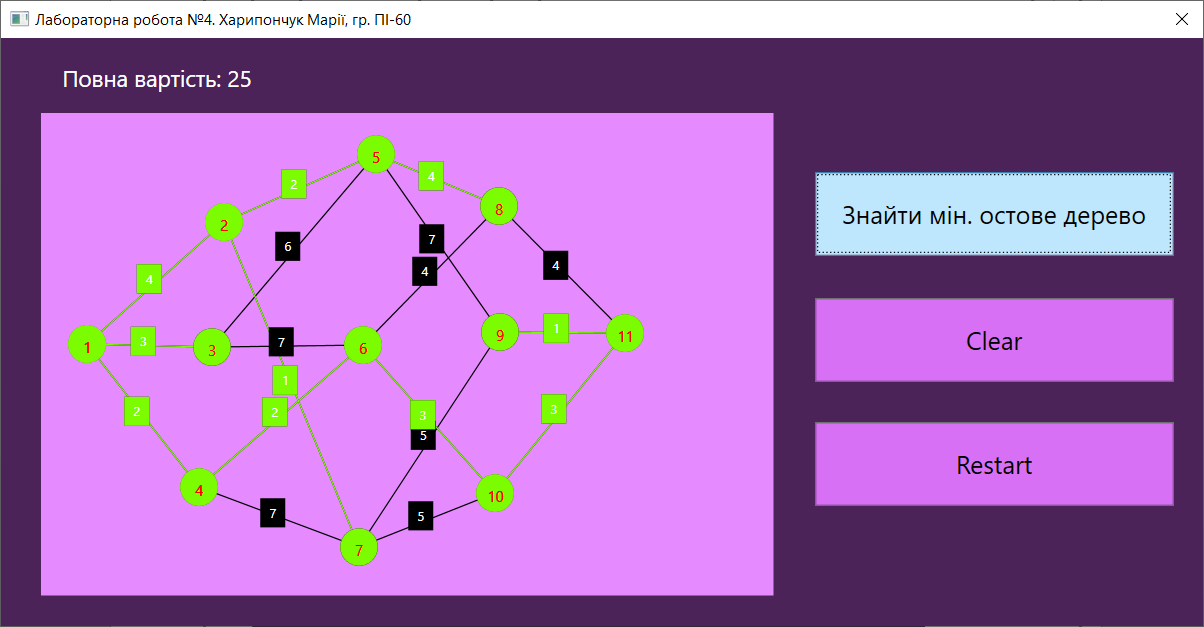


Рис 2.1 Приклад виконання програмного коду.

***Висновок*:** під час виконання даної лабораторної роботи було засвоєно основні операції над графами, знаходження мінімальної ваги за алгоритмом Пріма-Краскала.