

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»



Кафедра: САП

Звіт до виконаної лабораторної роботи №3
з дисципліни “Дискретна математика”
на тему: "Властивості графів. Операції над графами"

Виконав:

студент групи ПП-16

Якіб'юк Ігор

Прийняв:

Асистент каф. САП

Іванина В. В.

Львів - 2023

Лабораторна робота № 3

Мета роботи: Мета роботи – ознайомитись із можливостями системи Maple для вивчення властивостей графів, виконання операцій над графами та знаходження найкоротшого каркасу графа.

Хід роботи: Варіант 29

4.1:

Завдання 4.1. Властивості графів.

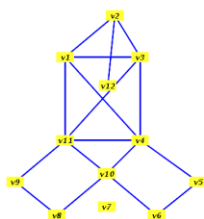
1. Зобразити в робочому вікні Maple18 заданий граф (Див. ДОДАТОК 1).
2. Чи є зв'язний заданий граф? Якщо ні, то “дорисуйте” його так, щоб він став зв'язним.
3. Побудуйте довільний граф та вкажіть за рисунком мости у ньому, якщо вони існують.

23

4. Побудуйте дводольні графи $K_{n,m}$ і $K_{p,l}$. Порахуйте кількість їх ребер і вершин.

Розв'язання:

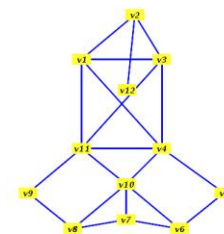
```
> with(GraphTheory):
> with(RandomGraphs):
> V := [v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12];
    V := [v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12]
> VP := [[-0.5, 1], [0.1, 1.5], [0.4, 1], [0.4, 0], [1.1, -0.5], [0.6, -0.9], [0, -0.8], [-0.6,
    -0.9], [-1.1, -0.5], [0, -0.4], [-0.5, 0], [0.01, 0.66]];
VP := [[-0.5, 1], [0.1, 1.5], [0.4, 1], [0.4, 0], [1.1, -0.5], [0.6, -0.9], [0, -0.8], [-0.6,
    -0.9], [-1.1, -0.5], [0, -0.4], [-0.5, 0], [0.01, 0.66]]
> G := Graph(V);
    G := Graph 25: an undirected unweighted graph with 12 vertices and 0 edge(s)
> SetVertexPositions(G, VP);
> AddEdge(G, {{v1, v2}, {v1, v4}, {v11, v3}, {v1, v3}, {v2, v3}, {v3, v4}, {v4, v5}, {v5,
    v6}, {v8, v9}, {v8, v10}, {v10, v4}, {v10, v6}, {v10, v11}, {v9, v11}, {v11, v4}, {v11,
    v1}, {v2, v12}});
    Graph 25: an undirected unweighted graph with 12 vertices and 17 edge(s)
> DrawGraph(G);
```



```
> IsConnected(G);
```

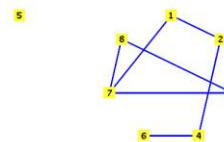
false

```
• AddEdge(G, {{v10, v7}, {v6, v7}, {v7, v8}});
    Graph 25: an undirected unweighted graph with 12 vertices and 20 edge(s)
• DrawGraph(G);
```

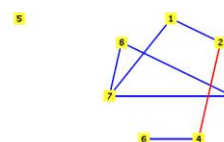


true

```
• IsConnected(G);
• F := RandomGraph(8, 0.3);
    F := Graph 26: an undirected unweighted graph with 8 vertices and 7 edge(s)
• DrawGraph(F);
```



```
• HighlightEdges(F, {2, 4});
• DrawGraph(F);
```

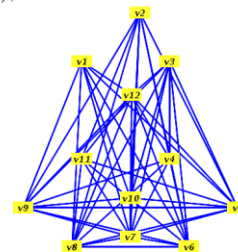
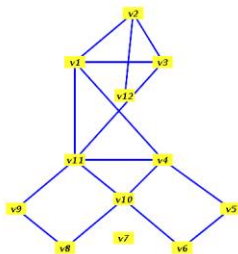
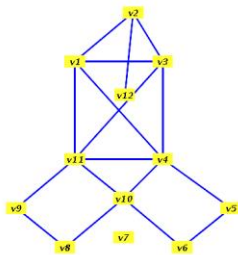


8

15

14

48

[illegible]

12 x 16 Matrix
Data Type: anything
Storage: sparse
Order: Fortran order

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 && 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

ShortestPath(*G*, *v5*, *v11*);

[*v5*, *v4*, *v11*]

Diameter(*G*);

Distance(*G*, *v5*, *v11*);

5

2

VP_A := [[0, 1], [1, 1], [0, 0], [1, 0], [0, -1], [1, -1]];

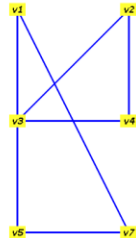
VP_A := [[0, 1], [1, 1], [0, 0], [1, 0], [0, -1], [1, -1]]

A := *Graph*({ {*v1*, *v3*}, {*v2*, *v3*}, {*v2*, *v4*}, {*v3*, *v4*}, {*v3*, *v5*}, {*v5*, *v7*}, {*v1*, *v7*} });

A := *Graph* 7: an undirected unweighted graph with 6 vertices and 7 edge(s)

SetVertexPositions(*A*, *VP_A*);

DrawGraph(*A*);

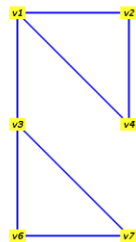


B := *Graph*({ {*v1*, *v2*}, {*v1*, *v3*}, {*v1*, *v4*}, {*v2*, *v4*}, {*v3*, *v7*}, {*v3*, *v6*}, {*v7*, *v6*} });

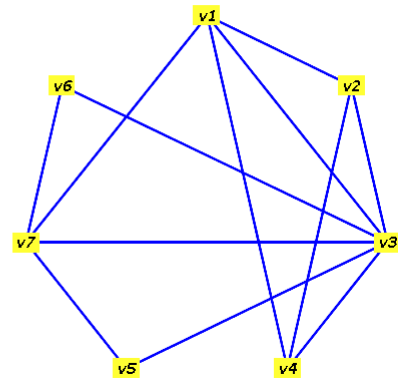
B := *Graph* 8: an undirected unweighted graph with 6 vertices and 7 edge(s)

SetVertexPositions(*B*, *VP_A*);

DrawGraph(*B*);



DrawGraph(*GraphUnion*(*A*, *B*));



Завдання 4.3:

Завдання 4.3. Знаходження найкоротшого кістякового дерева.

1. Задані у ДОДАТКУ 1 графи доповнити вагами ребер. Знайти найкоротше кістякове дерево заданого графа за алгоритмом Пріма.
2. Знайти найкоротше кістякове дерево заданого графа за алгоритмом Крускала з виведенням *N* його кроків.

Розв'язання:

V := [*v1*, *v2*, *v3*, *v4*, *v5*, *v6*, *v7*, *v8*, *v9*, *v10*, *v11*, *v12*];

V := [*v1*, *v2*, *v3*, *v4*, *v5*, *v6*, *v7*, *v8*, *v9*, *v10*, *v11*, *v12*]

VP := [[-0.5, 1], [0.1, 1.5], [0.4, 1], [0.4, 0], [1.1, -0.5], [0.6, -0.9], [0, -0.8], [-0.6, -0.9], [-1.1, -0.5], [0, -0.4], [-0.5, 0], [0.01, 0.66]];

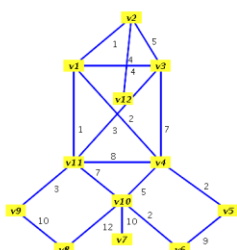
VP := [[-0.5, 1], [0.1, 1.5], [0.4, 1], [0.4, 0], [1.1, -0.5], [0.6, -0.9], [0, -0.8], [-0.6, -0.9], [-1.1, -0.5], [0, -0.4], [-0.5, 0], [0.01, 0.66]]

G := *Graph*(*V*, { { {*v1*, *v11*}, 1}, { {*v1*, *v2*}, 1}, { {*v1*, *v3*}, 4}, { {*v1*, *v4*}, 2}, { {*v10*, *v11*}, 7}, { {*v10*, *v4*}, 5}, { {*v10*, *v6*}, 2}, { {*v10*, *v7*}, 10}, { {*v10*, *v8*}, 12}, { {*v11*, *v3*}, 3}, { {*v11*, *v4*}, 8}, { {*v11*, *v9*}, 3}, { {*v12*, *v2*}, 4}, { {*v2*, *v3*}, 5}, { {*v3*, *v4*}, 7}, { {*v4*, *v5*}, 2}, { {*v5*, *v6*}, 9}, { {*v8*, *v9*}, 10} });

G := *Graph* 28: an undirected weighted graph with 12 vertices and 18 edge(s)

SetVertexPositions(*G*, *VP*);

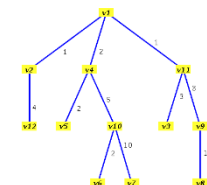
DrawGraph(*G*);



PrimaSpanTree := *PrimsAlgorithm*(*G*, *w'*);

PrimaSpanTree := *Graph* 29: an undirected weighted graph with 12 vertices and 11 edge(s)

DrawGraph(*PrimaSpanTree*);



infolevel[*KruskalsAlgorithm*] := 4

*infolevel*_{*GraphTheory*}:-*KruskalsAlgorithm* := 4

KruskalSpanTree := *KruskalsAlgorithm*(*G*, *w'*);

ruskalsAlgorithm: constructing minimal spanning tree on 12 vertices.

ruskalsAlgorithm: using Kruskal's algorithm at time: 0.812

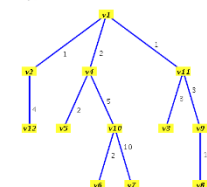
ruskalsAlgorithm: making heap of 18 edges at time: 0.812

ruskalsAlgorithm: finding the edges at time: 0.812

ruskalsAlgorithm: exiting Kruskal's algorithm at time 0.812

KruskalSpanTree := *Graph* 30: an undirected weighted graph with 12 vertices and 11 edge(s)

DrawGraph(*KruskalSpanTree*);



Висновок: На даній лабораторній роботі я навчився проводити різноманітні операції над графами в середовищі Maple18. Дослідив чи граф є зв'язний за допомогою команди `IsConnected()`. Побудував довільний граф і знайшов у ньому мости, а також побудував дводольні повні графи. Дізнався як і побудував матриці суміжності та інцидентності. При пошуку діаметра графа виникла проблема: метод `Diameter()` вертав безкінечність. Після успішної консультації з викладачем, проблема була вирішена і завдання виконане. Знайшов найкоротший шлях між вершинами у графі і їх кількість. В останньому завданні знайшов мінімальне каркасне дерево двома методами: Пріма і Крускала.