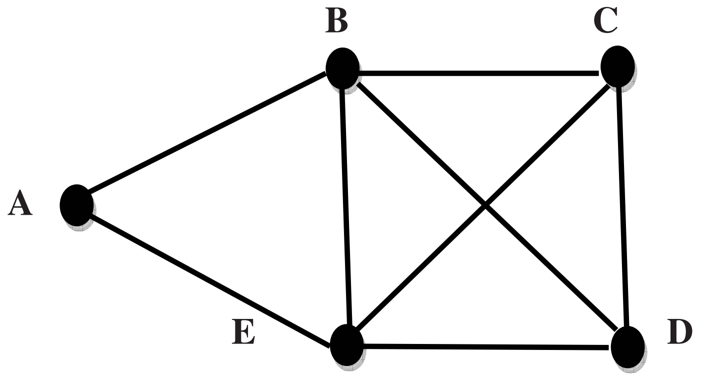
Tutorial Session 7 (Solution)

Exercise 1:

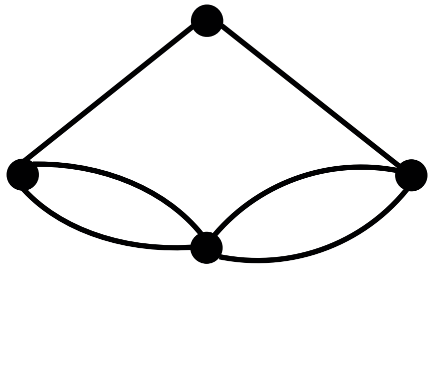
The following graph,



admits an Eulerian chain connecting vertices **C** and **D**, since the graph is connected and all vertices except **C** and **D** have even degrees. One possible Eulerian chain is: **(C, D, E, C, B, A, E, B, D).**

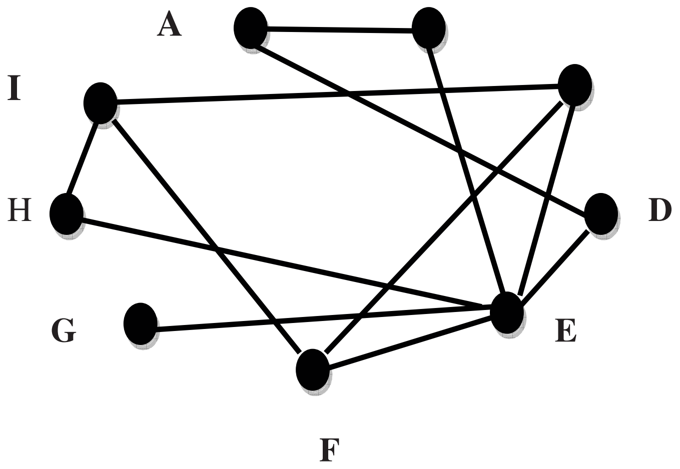
Exercise 2:

The Königsberg Bridge Problem has no solution, since the corresponding graph contains vertices of odd degree.



Exercise 3:

We begin by modeling the problem as a graph **G = (X , U)**. Each vertex **x** in **X** represents a country, and an edge **u** in **U** connects two vertices **x** and **y** if the corresponding countries are neighbors.



The goal is to colour the vertices so that no two adjacent vertices have the same colour.

The degrees of the vertices are:

$\mathbf{d}_G(\mathbf{A})=2,\ \mathbf{d}_G(\mathbf{B})=2,\ \mathbf{d}_G(\mathbf{C})=3,\ \mathbf{d}_G(\mathbf{D})=2,\ \mathbf{d}_G(\mathbf{E})=6,$

$\mathbf{d}_G(\mathbf{F})=3,\ \mathbf{d}_G(\mathbf{G})=1,\ \mathbf{d}_G(\mathbf{H})=2,\ \mathbf{d}_G(\mathbf{I})=3.$

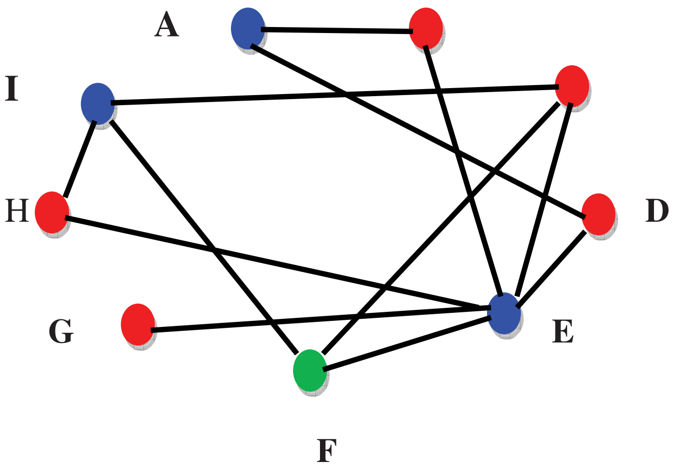
Vertices **E** has the maximum degree **6**. Assign **Colour 1** (**blue**) to vertex **E**.

After colouring vertex **E**, its neighbours **B**, **C**, **D**, **F**, **G**, **H** each have one coloured neighbour, hence their saturation degree is 1. To break the tie, we select the vertex with the highest degree. Choose vertex **C** (degree **3**). Its neighbour **E** already uses Colour 1; therefore, assign Colour 2 (**red**) to **C**.

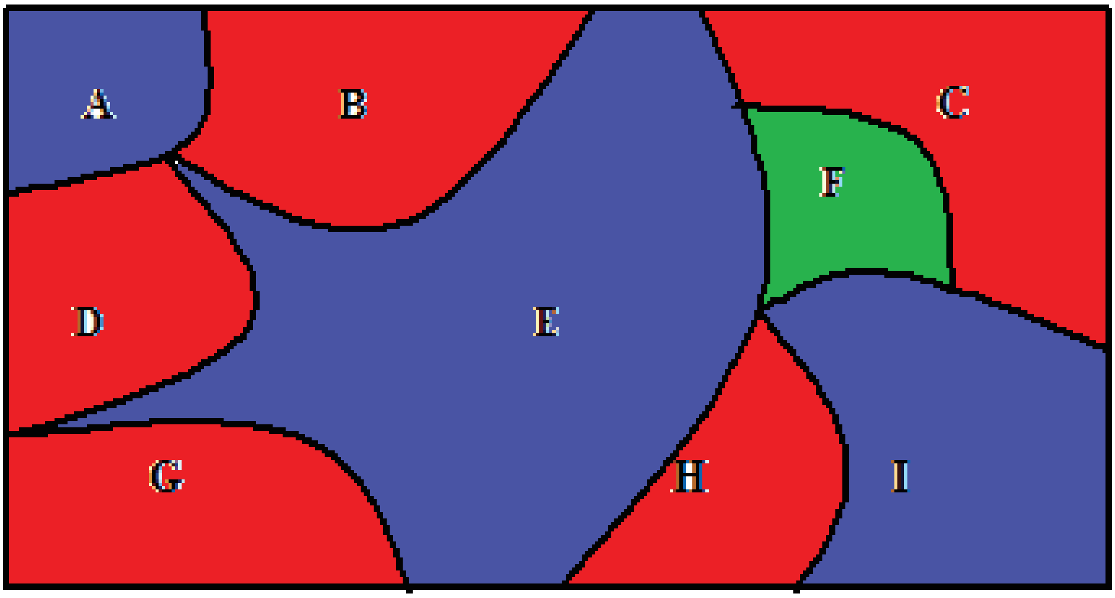
Vertex **F**, adjacent to both **E** (Colour 1) and **C** (Colour 2), has the highest saturation degree (2), thus assign Colour 3 (**green**) to vertex **F**.

Now, vertex **I** has the highest saturation degree (2). Its neighbours **C** and **F** use Colours 2 and 3 respectively, hence the smallest available colour is **Colour 1** (**blue**). Assign Colour 1 to vertex **I**.

Vertex **H** has the highest saturation degree (2). Its neighbours **E** and **I** use Colours 1, hence the smallest available colour is **Colour 2** (**red**). Assign **Colour 2** to vertex **H**. Using the same principle, we obtain the final coloring of both the graph and the map.

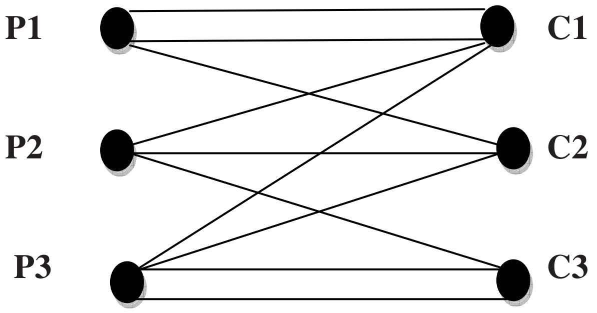


Thus, the *chromatic number* $\gamma(\mathbf{G}) \leq 3$.



Exercise 4:

We model the problem using a ***bipartite graph*** with two sets of vertices: **teachers (P)** and **classes (C)**. Each edge connects a teacher to a class, representing a teaching hour.



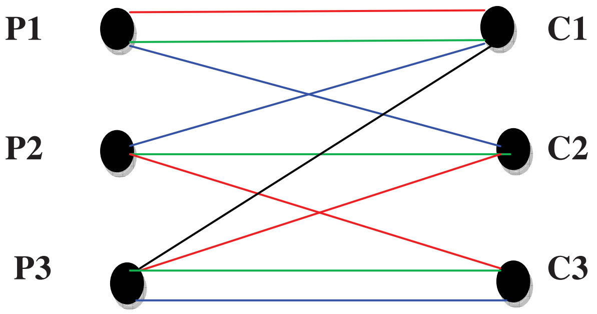
The **maximum number of time slots** needed equals the **maximum number of teaching hours** for any teacher, which corresponds to the **maximum degree** among the teacher vertices.

Since **Max(d(P_i)) = d(P_3) = 4**, we need **4 time slots** in total.

We use the following color convention for the schedule:

- 1st hour → **red**
- 2nd hour → **green**
- 3rd hour → **blue**
- 4th hour → **black**

The coloring is assigned step by step, starting with the first hour, then the second, third, and fourth, as shown in the final graph.


P1
P2
P3
C1
C2
C3