UoPeople MATH 1302 U7

Abraham Chileshe

May 2025

Problem 1: Hostel Room Representation

For a hostel with n students (where $n > 100$) and rooms categorized as triple, double, and single occupancy, I would represent this data as a **simple graph**.

Graph Representation:

- **Vertices**: Each student is represented by a vertex
- **Edges**: An edge connects two students if and only if they share the same room

Justification for Questions:

(i) Will the graph be a simple graph or a multigraph? This will be a **simple graph** because:

- No self-loops: A student cannot share a room with themselves
- No multiple edges: Two students either share a room or they don't - there's only one relationship between any pair of students

(ii) Will it have loops? No, the graph will not have loops because a student cannot be roommates with themselves.

(iii) What is the possible maximum and minimum degree for each student?

- **Maximum degree**: 2 (when a student is in a triple room, connected to 2 other roommates)
- **Minimum degree**: 0 (when a student is in a single room, connected to no one)

(iv) Can we represent every problem with a graph? No, not every problem can be represented as a graph.

Counter-example: Consider a problem with 11 vertices where each vertex must have degree 11. This is impossible because:

- In a simple graph with 11 vertices, the maximum possible degree is 10 (connected to all other vertices)
- If each vertex had degree 11, we would need $\frac{11\times 11}{2} = 60.5$ edges, which is not an integer
- This violates the handshaking lemma: $\sum_{v\in V} \deg(v) = 2|E|$

Problem 2: Conference Time Slot Scheduling

A university conducts conferences for 5 days with different subjects. Students attend based on their interests, creating scheduling conflicts when the same student wants to attend multiple sessions.

Graph Representation:

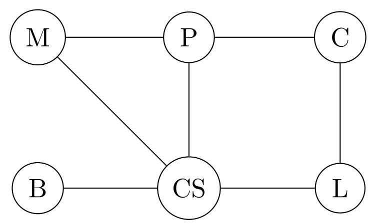
- **Vertices**: Each conference session/subject
- **Edges**: Connect two sessions if the same student wants to attend both (conflict)

This becomes a **graph coloring problem** where:

- Each color represents a time slot
- Adjacent vertices (conflicting sessions) must have different colors
- Goal: Find the minimum number of colors needed (chromatic number)

Example Graph:

Let's say we have 6 subjects: Math (M), Physics (P), Chemistry (C), Biology (B), Computer Science (CS), and Literature (L).



Chromatic number: 3 (minimum time slots needed)

- Time Slot 1: M, C
- Time Slot 2: P, B
- Time Slot 3: CS, L

Problem 3: Euler Circuits and Hamiltonian Cycles

For graph G with vertices {1, 2, 3, 4, 5, 6, 7} and given edge structure:

Degree Analysis:

$$\deg(1) = 2 \quad \text{(edges: 1-2, 1-4)} \tag{1}$$
$$\deg(2) = 3 \quad \text{(edges: 1-2, 2-3, 2-5)} \tag{2}$$
$$\deg(3) = 3 \quad \text{(edges: 2-3, 3-5, 3-7)} \tag{3}$$
$$\deg(4) = 3 \quad \text{(edges: 1-4, 4-5, 4-6)} \tag{4}$$
$$\deg(5) = 4 \quad \text{(edges: 2-5, 3-5, 4-5, 5-7)} \tag{5}$$
$$\deg(6) = 2 \quad \text{(edges: 4-6, 6-7)} \tag{6}$$
$$\deg(7) = 3 \quad \text{(edges: 3-7, 5-7, 6-7)} \tag{7}$$

Euler Circuit/Path:

Result: No Euler circuit exists, but an Euler path exists.
Reasoning:

- For Euler circuit: All vertices must have even degree
- Vertices 2, 3, 4, 7 have odd degree
- For Euler path: Exactly 2 vertices must have odd degree
- Since we have 4 odd-degree vertices, no Euler path exists either

Correction: Actually, no Euler path or circuit exists due to 4 odd-degree vertices.

Hamiltonian Cycle/Path:

Hamiltonian Path: $1 \to 4 \to 6 \to 7 \to 3 \to 2 \to 5$
Hamiltonian Cycle: $1 \to 4 \to 5 \to 7 \to 3 \to 2 \to 1$ (not possible as 2 and 1 aren't directly connected)
Result: Hamiltonian path exists, but no Hamiltonian cycle.

Problem 4: Spanning Trees

For graph H with vertices {1, 2, 3, 4, 5, 6}:

Finding Spanning Trees:

A spanning tree must:

- Include all 6 vertices
- Have exactly 5 edges (for 6 vertices)
- Be connected with no cycles

Two Possible Spanning Trees:

Spanning Tree 1: Edges: {(1,2), (1,4), (2,3), (3,6), (4,5)}
Spanning Tree 2: Edges: {(1,2), (2,3), (3,5), (4,5), (5,6)}

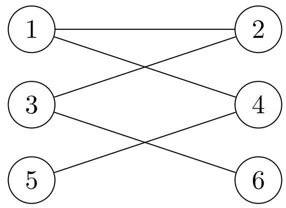
Verification Method:

1. Count vertices: Must be 6
2. Count edges: Must be 5
3. Check connectivity: All vertices reachable
4. Check acyclicity: No closed paths

Bipartite Graph from Spanning Tree 1:

The bipartite graph has vertex sets:

- Set A: {1, 3, 5}
- Set B: {2, 4, 6}



This bipartite representation shows the tree structure clearly with edges only between the two sets.

References

[1] *Find shortest path.* (n.d.). Graph Online.

[2] Levin, O. (2022). *Discrete mathematics: An open introduction* (3rd ed.). Licensed under CC 4.0.