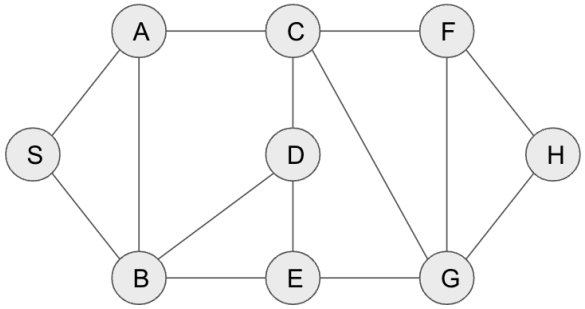
Assignment 3 Semester: Summer 2025
Deadline: 01/09/25, 11:59 PM Full Marks: 30

CSE 221: Algorithms

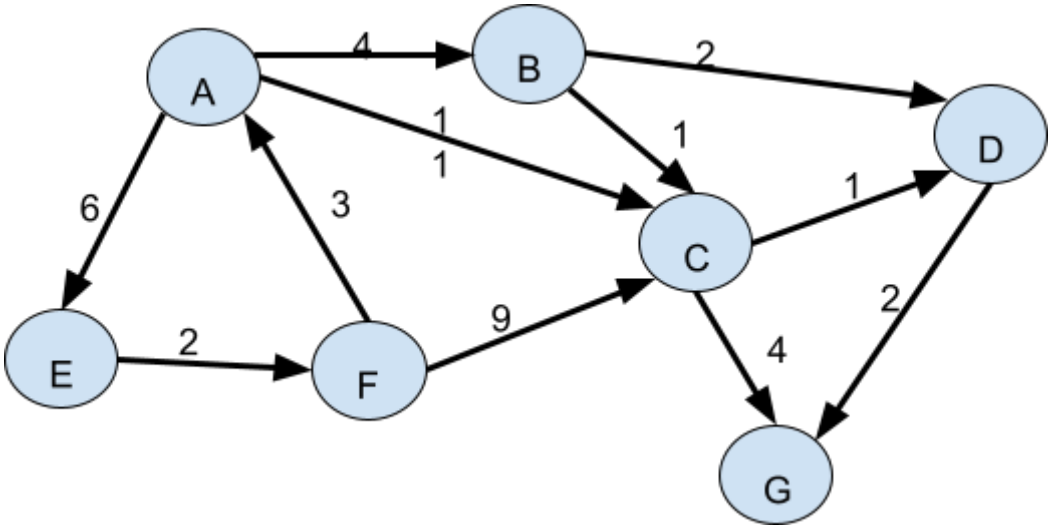
1. Consider this graph:



a) Justify $\sum_{v \in V} deg(v) = 2m$, where m = number of edges in the graph. [3]

b) Calculate the number of additional edges that can be added between the nodes randomly, keeping the graph 'simple' (without adding multiple edges between any two nodes, loops in a single node).

2.



a. Now given that you are in city E and want to find the shortest path to every other city. Now apply a suitable algorithm to find your desired output and show all the steps as well.

b. Now consider that the roads that connect the cities have become bidirectional and connect all the cities with minimum distance and show the road connections. Now apply a suitable algorithm to find the road connection.

3. You have just joined a new university and are exploring the campus, which consists of several buildings connected by one-way bridges. These bridges form a directed graph, where each node represents a building and each edge represents a bridge between two

buildings. You aim to analyze how the buildings and bridges are connected to understand the campus layout better and improve your navigation between classes and facilities.

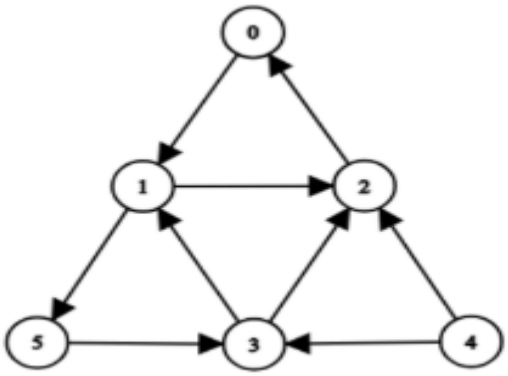


Figure 1

a) Mention which graph representation will let you check whether any two buildings have a bridge between them in O(1) time.

b) From Figure 1

i. Simulate an algorithm to find the Strongly Connected Components (SCC).

ii. Suppose a bridge gets blocked. Explain how you can efficiently calculate the updated SCCs from b(i) without rerunning the algorithm on the entire graph.

c) A building is called “Accessible” if it's possible to start from there, visit some (or all) of the other buildings, and eventually return to that building.
Propose an algorithm with a pseudocode/programmable code/step-by-step logical instructions to determine whether every building on the campus is Accessible. The algorithm should run in at most O(V × (V + E)) time.

4. The country of Technovia is setting up an emergency grid that connects 8 major cities: A (Capital), B, C,D, E, F, G, H. These cities are connected by one-way roads. Each road has a traffic weight which is used to estimate the delivery delay due to congestion. Medical aid and personnel must be delivered from the capital (A) to all other cities. The map of the country is given below:

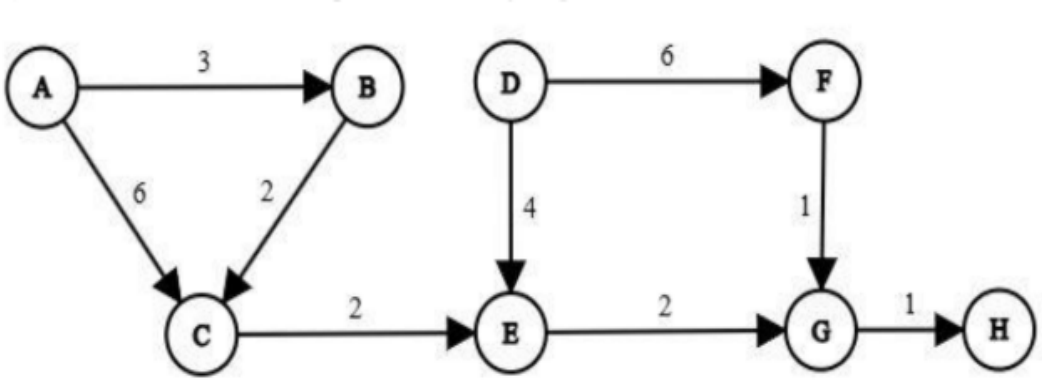


Figure 2: Q2.

a. Suppose you, an algorithm enthusiast, have been assigned to determine the minimum delivery time from capital A to all other cities.

1. Which SSSP (single source shortest path) algorithm would you prefer to solve the problem?
2. Simulate your mentioned algorithm with the necessary data structures to find the minimum delivery times from the capital to all other cities. Travel times are equal to the traffic weights of the roads.

b. Continuing from the given scenario, the Ministry of Transport enforces that while delivering medical resources, after using any high-traffic road (traffic weight > 4), the next road used in the delivery path must have a traffic weight ≤ 4. This is to prevent back-to-back gridlock. Propose a modification to your algorithm of Q2(a) to determine the minimum delivery time from capital A to all other cities while strictly following the additional constraint of traffic weights. Present your idea using programmable code/pseudocode/step-by-step instructions.

c. Now the Ministry wants to design a minimum-traffic infrastructure plan that connects all the cities (an MST of the graph). Will Kruskal's or Prim's MST finding algorithm work properly for the given one-way roads? If yes, justify your answer. Otherwise, propose necessary modifications to the graph so that the algorithm works.

5. A data analytics company is designing a batch processing system consisting of eight tasks, labeled T1 to T8. Due to data and resource dependencies, the following constraints apply:

- T1 must be completed before T3 and T4
- T4 must be completed before T6 and T7
- T2 must be completed before T4 and T5
- T6 must be completed before T8
- T7 must be completed before T8
- T3 must be completed before T6
- T5 must be completed before T7

The system architect wants to determine a feasible order in which all tasks can be executed.

(a) Draw or describe the directed graph representing the task dependencies.

(c) Now using a suitable algorithm, determine one valid task execution sequence.

(d) Identify whether multiple valid schedules are possible. Justify your answer based on the dependency structure.

(e) Suppose an additional dependency T8 → T2 is introduced. Explain how this affects the task schedule without recomputing the entire order.

6. A large-scale distributed system consists of **N interconnected computing nodes**, modeled as a **directed graph** (G = (V, E)). Each vertex represents a node, and a directed edge represents a **one-way communication channel**.

1 : 2, 4

2 : 3, 5

3 : 1, 6

4 : 5

5 : 6, 7

6 : 4, 8

7 : 8, 9

8 : 10

9 : 7

10 : ∅

Due to partial system failures and security constraints, the following conditions apply:

1. The graph may contain **cycles**, **self-loops**, and **disconnected components**.
2. Some nodes are marked as **critical nodes**. Once a critical node is visited, all of its outgoing edges must be explored **before any other node at the same traversal depth** is processed.
3. The traversal must:
    - Visit **every reachable node** from a given source (S)

    - Detect whether a **cycle is reachable from (S)**
    - Classify each reachable edge as **tree, back, forward, or cross edge**
4. The traversal must minimize **maximum memory usage**, as the system operates under strict memory constraints.

A systems engineer must choose an appropriate **graph traversal strategy** and justify its correctness and efficiency under these constraints.

(a) Among **Breadth-First Search (BFS)** and **Depth-First Search (DFS)**, identify which traversal algorithm is **more suitable** for the given scenario. Justify your choice and simulate with respect to:

- Cycle detection
- Edge classification
- Memory constraints

(b) Explain how the selected traversal algorithm can be **modified** to enforce the rule that **all outgoing edges of a critical node must be explored before other nodes at the same traversal depth**. Clearly state any additional data structures or ordering constraints required.

(c) During traversal, a node (u) is discovered at time (d[u]) and finished at time (f[u]).
Explain how these timestamps can be used to:

- Detect cycles reachable from the source
- Distinguish between **back edges** and **cross edges**

(d) Suppose the system switches from a **directed graph** to an **undirected graph** while keeping all other constraints unchanged.
Analyze how this change affects:

- Cycle detection
- Edge classification
- Correctness of the traversal strategy

(e) Prove or disprove the following statement:

> "If a graph traversal starting from (S) visits all reachable nodes and detects no back edges, then the reachable subgraph is acyclic."

Your answer must include a **formal argument or counterexample**.

(f) Let (V_r) and (E_r) denote the number of vertices and edges reachable from (S).
Derive the **time and space complexity** of the traversal algorithm under worst-case conditions, and explain how the imposed constraints affect these bounds.