Due Date: 30 November 2025
20% penalty for 1 day late
40% penalty for 2 days late

CS2009: Design and Analysis of Algorithms (Fall 2025)
Assignment 4

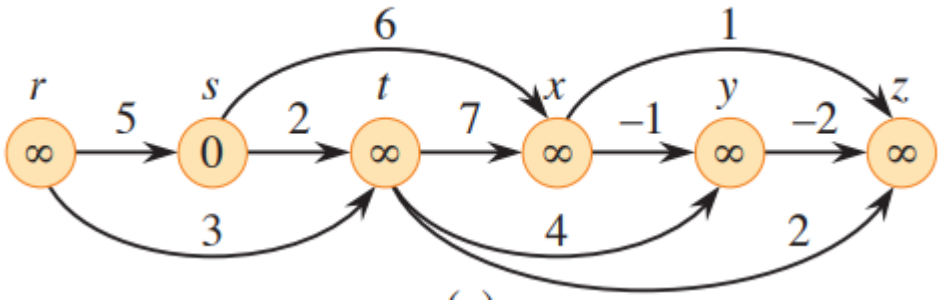
Submission not allowed afterwards

Total Marks: 100

1. Given a weighted, directed graph $G = (V, E)$ with no negative-weight cycles, let m be the maximum over all vertices $v \in V$ of the minimum number of edges in a shortest path from the source s to v. (Here, the shortest path is by weight, not the number of edges.) Suggest a simple change to the Bellman-Ford algorithm that allows it to terminate in $m + 1$ passes, even if m is not known in advance. [10 Marks]

2. The following algorithm finds single source shortest paths in DAG. Dry run the following algorithm for finding shortest paths to all nodes, for the following graph, start from node r. Analyze the algorithm and provide its complexity in big O notations. [10 Marks]

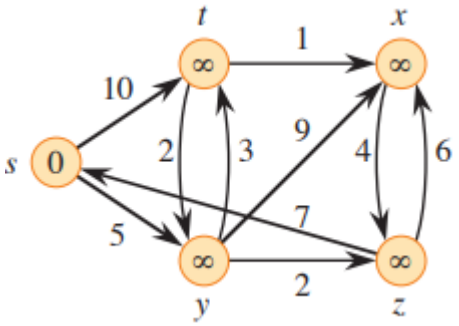
```
DAG-SHORTEST-PATHS(G, w, s)
1  topologically sort the vertices of G
2  INITIALIZE-SINGLE-SOURCE(G, s)
3  for each vertex u ∈ G.V, taken in topologically sorted order
4      for each vertex v in G.Adj[u]
5          RELAX(u, v, w)
```



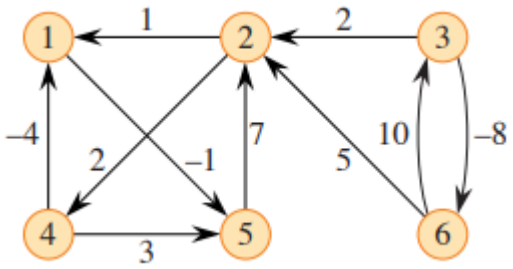
3. Run Dijkstra's algorithm on the directed graph given below, using vertex s as the source and then using vertex z as the source. Show the d and π values and the vertices in set S after each iteration of the while loop. [10 Marks]

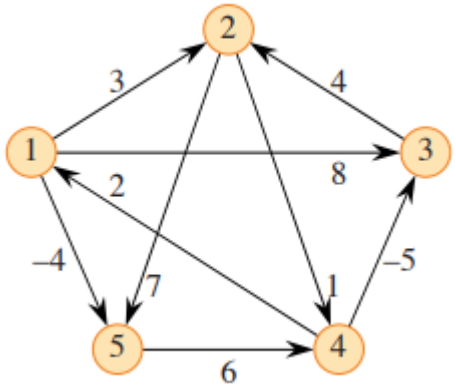
Submission not allowed afterwards
Total Marks: 100

```
DIJKSTRA(G, w, s)
1  INITIALIZE-SINGLE-SOURCE(G, s)
2  S = ∅
3  Q = ∅
4  for each vertex u ∈ G.V
5      INSERT(Q, u)
6  while Q ≠ ∅
7      u = EXTRACT-MIN(Q)
8      S = S ∪ {u}
9      for each vertex v in G.Adj[u]
10         RELAX(u, v, w)
11         if the call of RELAX decreased v.d
12             DECREASE-KEY(Q, v, v.d)
```



4. Use Floyd-Warhall Algorithm to find all pair shortest paths in the following graphs. Specify all the path. [10 Marks]





5. a) Conduct a DFS for the following graph. (Label each vertex u with the start time and the finish time) or (Show all steps showing stack and Visit Order). You should start the traversal from vertex B, and follow the alphabetic order whenever you need to make choices. [10 Points]

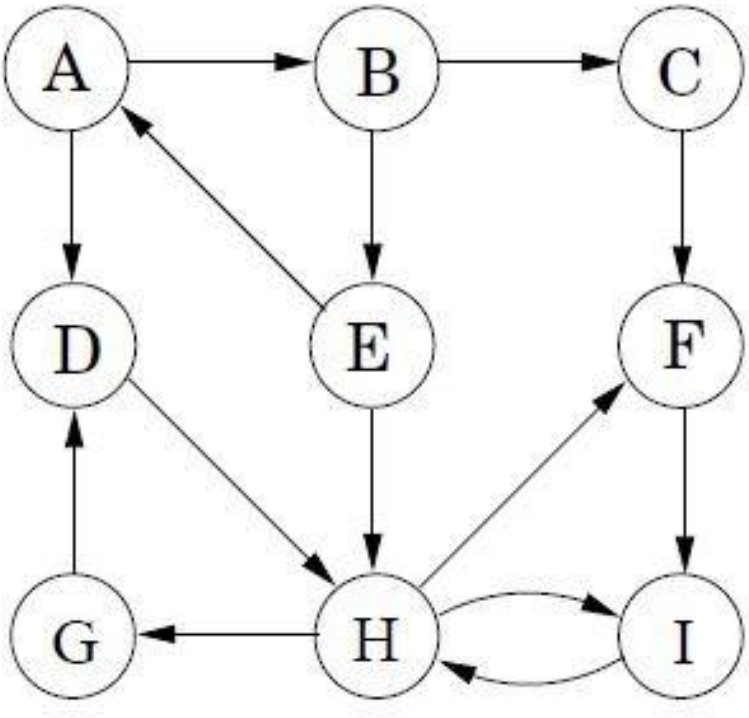


Figure 1: Graph for the DFS and SCC

b) List all edges (back edges, forward edges, cross edges) that belong to each of the following sets: [5 Marks]

c) Identify the strongly connected components and draw the component graph. [10 Marks]

Submission not allowed afterwards
Total Marks: 100

6. Use Prim's algorithm to compute a maximum spanning tree for the following graph shown in figure 2. Use node A as the root. You need to show each step. [10 Marks]

   b) Can Prim's and Kruskal's algorithm yield different maximum spanning trees? Explain why or why not. [5 Marks]

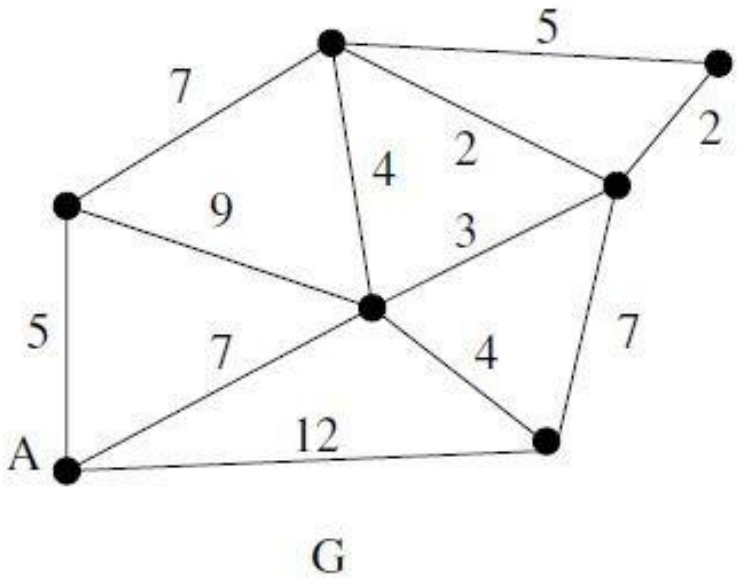


Figure 2: Graph for a maximum spanning tree

7. a) Give a simple example of a directed graph with negative-weight edges for which Dijkstra's algorithm produces incorrect answers. [10 Marks]

   Professor A suggests that to tackle negative-weight edges issue of Dijkstra's algorithm: add a large constant to each edge weight so that all the weights become positive, then run algorithm starting at node s, and return the shortest path found to node t.

   b) Is this a valid method? Either prove that it works correctly, or give a counter example. [5 Marks]

   c) Consider a directed graph in which the only negative edges are those that leave s; all other edges are positive. Can Dijkstra's algorithm, started at s, fail on such a graph? Prove your answer. [5 Marks]