Due Date: 27 November 2022
20% penalty for 1 day late
40% penalty for 2 days late
Submission not allowed afterwards

CS2009: Design and Analysis of Algorithms (Fall 2022)
Assignment 4
Total Marks: 100

1. 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 A, and follow the alphabetic order whenever you need to make choices. [10 Points]

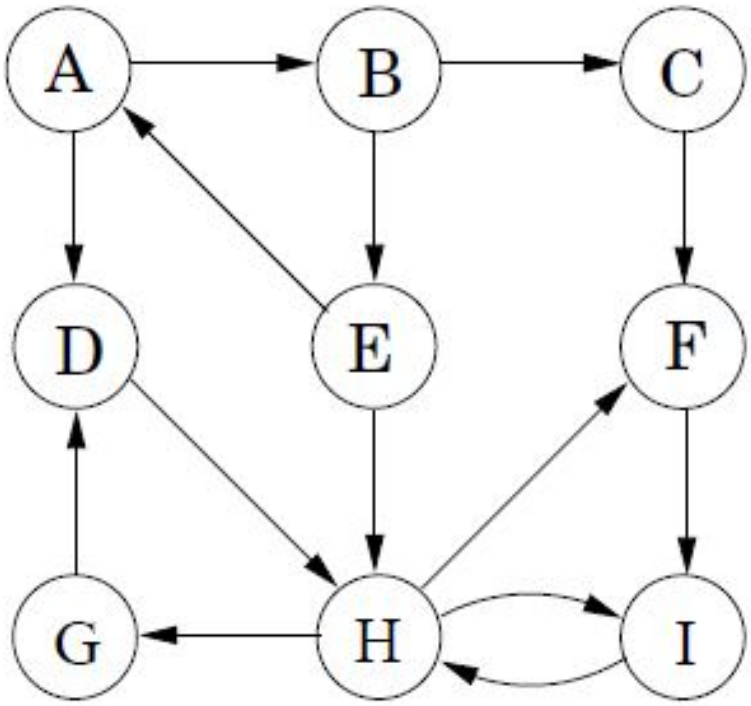


Figure 1: Graph for the DFS and SCC

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

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

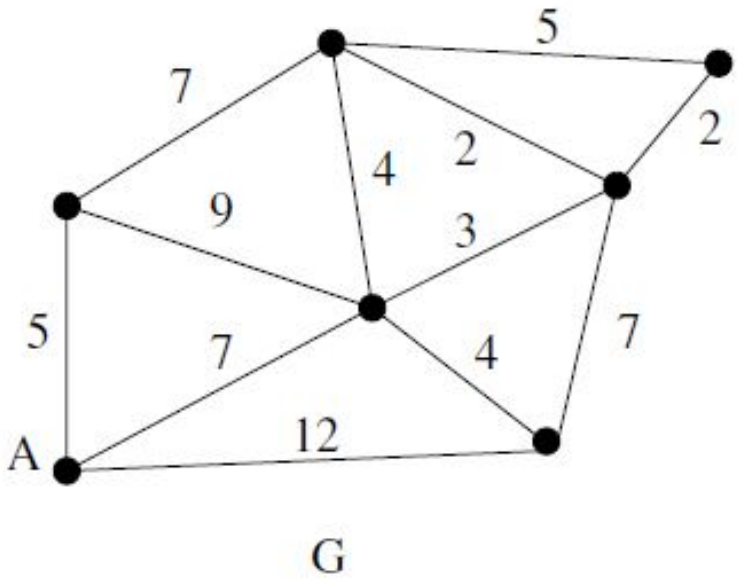
2. A bipartite graph is a graph G = (V, E) whose vertices can be partitioned into two sets ($V = V_1 \cup V_2$ and $V_1 \cap V_2 = \phi$) such that there are no edges between vertices in the same set (for instance, if $u, v \in V_1$, then there is no edge between u and v).

   a) Design a linear-time algorithm to determine whether an undirected graph is bipartite. [10 points]

   b) At most how many colors are needed to color in an undirected graph with exactly one odd length cycle? [5 points]

3. 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 Points]

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



G

Figure 2: Graph for a maximum spanning tree

4. Using Floyd-Warshall, find all pairs shortest path for the given Figure 3 (D^0 weight matrix is also provided). Discuss its time and space complexity as well [10 Points]

5. Go through the lecture of Johnson's Algorithm part 1, Johnson's Algorithm part 2 and write its summary. [10 Points]

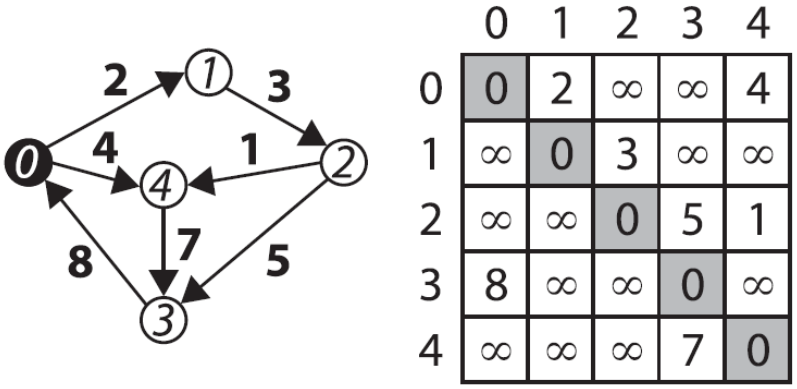
6. a) Give a simple example of a directed graph with negative-weight edges for which Dijkstra's algorithm produces incorrect answers.(5 points)

   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 counterexample.(5 points)

   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 Points]

7. Read Book, Lectures or Search through Internet to explain the worst time complexities of BFS, DFS, Kruskal's and Prims in your own words using (i) Adjacent Matrix (ii) Adjacent List [10 Points]



	0	1	2	3	4
0	0	2	∞	∞	4
1	∞	0	3	∞	∞
2	∞	∞	0	5	1
3	8	∞	∞	0	∞
4	∞	∞	∞	7	0

Figure 3: find all pairs shortest path