Task 1 (a)

This problem required the direct implementation of Topological sort. Topological sort can't be implemented in a graph where cycle exists. That's why while doing Topsort using DFS, checked for the cycle in the graph.

Task 1 (b)

This problem is about also topsort but using BFS. While doing BFS used the knowledge to indegree. After that visited node stored in an array if arrays length and total node count is not same, thus the graph has a cycle. which means it's not a DAG (Directed Acyclic Graph). Thus topsort is impossible.

Task (2)

For task 2 used priority queue. Using priority queue in the BFS technique of topsort solve the problem that asked here

Task (3)

Used 4 steps to find (SCC) strongly connected component. In the first step and second step we do normal DFS and find the nodes like topsort (not topsort actually)

Then we do DFS on the transpose graph in the sequence of Step 2.

This algorithm is called KOSARAJU.