

Task 1(a): This task works on finding sequence based on conditions through DFS approach. Here we visited each node and pushed the node while traversing in the stack, and once its all neighbours were explored, the node was popped and pushed into an container - that in its reversed order is our output. Anytime a new node was explored, it was pushed into the stack.

Task 1(b): This task works on finding sequence based on conditions through BFS approach. Initially the indegrees were stored for corresponding nodes. If any node had an indegree of '0', it was pushed into a queue. Each time the first elem of the queue was explored based on its neighbours and their in degree were reduced by 1 until the time when it reached 0 and it/node was enqueued to the queue.

Once exploration was complete, the queue popped the element to our answer container and this process continued till our queue turned empty.

Task 2: To find the lexicographically smaller graph we took the help of BFS approach. It is similar to the idea of Task 1(b) but instead of queue, we introduced the idea of "min heap" that chose the minimum element of "min" each time to explore its neighbour..

Task 3: Here, in order to find the SCC we took the help of kosaraju algorithm while following dfs. We called dfs for the main graph and again for that of its transposed graph. This helped to find out the strongly connected components within the list lastly.