1a. Wrote DFS code and checked if there is any cycle within the graph .If there is, then printed 'IMPOSSIBLE' else printed the nodes at descending order of their finish time.

1b. Used adjacency matrix and counted indegree of every node. Check if there is any cycle. If there is not, then print nodes with 0 indegree and reduce indegree by 1 of all adjacent nodes. By repeating this, we get a topological sort.

2. Used BFS to get the topological sort. When dequeuing a node, First sorted the elements of the queue then dequeued. So. BFS is done lexicographically and topological sort is done in that order.

3. Ran DFS on the graph. Transposed the graph. Then ran DFS in the transpose graph in the previous dfs order. Then printed the isolated DFS result and got the Strongly connected components