Sub.:

SAT SUN MON TUE WED TH

DATE: / /

Task-1A

I used the idea of topological sort to Solve the problem. first i converted the input file into adjacent list. I stored all courses with prerequisite in a different set and then find out the difference. By these way I figured out which courses doesnot have prerequisites. Then I ran dfs on them and stored done one in result array. then I checked whether there's a cycle and printed 'Impossible'

1b

I used the idea of kahn's Algorith (using Bfs) to solve problem. ~~fo~~ I created a dictionary to store how many prerequisite that cours have. I create a queue with courses ~~has~~ that has no prerequisite. then I dequeue one by one and stored in result array and reduce indegree for all its adjacent nodes. lastly I printed 'Impossible' if the result array has more / less course then inputs.

Sub.: DATE: / /

## Task-2

I used the idea of khan's Algorithm (using BFS) to solve problem. I creat dictionary to store how many prerequisite that course have. I create a queue ~~with~~ that has no prerequis then I sorted the queue to find the lexicographically smallest valid course sequence. i dequeue one by one and stored then in result array.

## Task-3

I perform DFS and push complete nodes in a stack. I transposed the graph I again perform dfs but these time I poped items from stack and stored visited nodes in array and stored that array in another list. I continue until stack is empty the array in result are strongly connecte components