**Topological Sort: Any linear ordering of vertices such that if there is an edge between u&v, then u appears before v. It is only possible in Directed acyclic graph. This algorithm is used to solve the dependency between the modules and that is the reason it is only applicable for DAG.**

**It can be possible to do this sort using DFS and BFS.**

**For DFS, we have to use stack and visited array. And in case of BFS, we should use indegree array and queue.**

Converting the vector<vector<int>> to vector<int> adj[];

vector<vector<int>>&prerequisites;

vector<int> adj[numCourses]

for(inti=0;i<prerequisites.size();i++){

adj[prerequisites[i][1]].push\_back(prerequisites[i][0]);

}