210. Course Schedule II

vector<int> findOrder(int numCourses, vector<vector<int>>& prerequisites) {

// build graph first

// graph[pre0][pre1] --> graph[pre0] = {pre1, pre2, ...}

vector<vector<int>> graph(numCourses);

for(auto pre : prerequisites)

graph[pre[1]].push\_back(pre[0]);

// 0 : unvisit

// 1 : visiting

// 2 : visited complete

vector<int> visited(numCourses, 0);

vector<int> res;

for(int i = 0; i < numCourses; i++){

if(visited[i] == 0 && !dfs(graph, visited, i, res)) return {};//false;

}

reverse(res.begin(), res.end());

return res;//true;

}

bool dfs(vector<vector<int>>& graph, vector<int>& visited, int v, vector<int>& res){

if(visited[v] == 1) return false;

if(visited[v] == 2) return true;

visited[v] = 1;

for(auto& neighbor : graph[v]){

if(!dfs(graph, visited, neighbor, res)) return false;

}

visited[v] = 2;

res.push\_back(v);

cout << v << endl;

return true;

}