207. Course Schedule <Medium>

<DFS> <topological Sort>

class Solution {

public:

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

// 0 : unvisit

// 1 : visiting

// 2 : visited complete

vector<int> visited(numCourses, 0);

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

if(!dfs(graph, visited, i)) return false;

}

return true;

}

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

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

visited[v] = 1;

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

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

}

visited[v] = 2;

return true;

}

};