**SOLUTION**

class Solution {

public:

bool canFinish(int numCourses, vector<vector<int>>& prerequisites) {

vector<vector<int>> adj(numCourses,vector<int>());

for(vector<int>& pre: prerequisites)

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

vector<int> visited(numCourses,0);

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

if(visited[i] == 0 && !dfs(adj,visited,i))

return false;

return true;

}

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

if(visited[v] == 1)

return false;

visited[v] = 1;

for(int ad:adj[v])

if(!dfs(adj,visited,ad))

return false;

visited[v] = 2;

return true;

}

};

**TIME COMPLEXITY= O(N)**

**SPACE COMPLEXITY= O(N)**