</> Code

Java Auto

```java
class Solution {
    public boolean canFinish(int numCourses, int[][] prerequisites) {
        List<List<Integer>> adj = new ArrayList<>();
        for (int i = 0; i < numCourses; i++) {
            adj.add(new ArrayList<>());
        }
        for (int[] pair : prerequisites) {
            adj.get(pair[0]).add(pair[1]);
        }
       int[] visited = new int[numCourses];
        for (int i = 0; i < numCourses; i++) {
            if (!dfs(i, adj, visited)) {
                return false;
            }
        }
     return true;
    }
   private boolean dfs(int course, List<List<Integer>> adj, int[] visited) {
        if (visited[course] == 1) {
            return false;
        }
        if (visited[course] == 2) {
            return true;
```

Saved

```java
                return false;
            }
        }
     return true;
    }
   private boolean dfs(int course, List<List<Integer>> adj, int[] visited) {
        if (visited[course] == 1) {
            return false;
        }
        if (visited[course] == 2) {
            return true;
        }
        visited[course] = 1;
        for (int neighbor : adj.get(course)) {
            if (!dfs(neighbor, adj, visited)) {
                return false;
            }
        }
      visited[course] = 2;
        return true;
    }
}
```