Problem 207: Course Schedule

Problem Information

Difficulty: Medium
Acceptance Rate: 50.28%
Paid Only: No
Tags: Depth-First Search, Breadth-First Search, Graph, Topological Sort

Problem Description

There are a total of `numCourses` courses you have to take, labeled from `0` to `numCourses - 1`. You are given an array `prerequisites` where `prerequisites[i] = [ai, bi]` indicates that you **must** take course `bi` first if you want to take course `ai`.

* For example, the pair `[0, 1]`, indicates that to take course `0` you have to first take course `1`.

Return `true` if you can finish all courses. Otherwise, return `false`.

Example 1:

Input: numCourses = 2, prerequisites = [[1,0]] **Output:** true **Explanation:** There are a total of 2 courses to take. To take course 1 you should have finished course 0. So it is possible.

Example 2:

Input: numCourses = 2, prerequisites = [[1,0],[0,1]] **Output:** false **Explanation:** There are a total of 2 courses to take. To take course 1 you should have finished course 0, and to take course 0 you should also have finished course 1. So it is impossible.

Constraints:

* `1 <= numCourses <= 2000` * `0 <= prerequisites.length <= 5000` * `prerequisites[i].length == 2` * `0 <= ai, bi < numCourses` * All the pairs prerequisites[i] are **unique**.

Code Snippets

C++:

```
class Solution {
public:
bool canFinish(int numCourses, vector<vector<int>>& prerequisites) {

}
};
```

Java:

```
class Solution {
public boolean canFinish(int numCourses, int[][] prerequisites) {

}
}
```

Python3:

```
class Solution:
def canFinish(self, numCourses: int, prerequisites: List[List[int]]) -> bool:
```