Problem 210: Course Schedule II

Problem Information

Difficulty: Medium
Acceptance Rate: 54.48%
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 _the ordering of courses you should take to finish all courses_. If there are many valid answers, return **any** of them. If it is impossible to finish all courses, return **an empty array**.

Example 1:

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

Example 2:

Input: numCourses = 4, prerequisites = [[1,0],[2,0],[3,1],[3,2]] **Output:** [0,2,1,3] **Explanation:** There are a total of 4 courses to take. To take course 3 you should have finished both courses 1 and 2. Both courses 1 and 2 should be taken after you finished course 0. So one correct course order is [0,1,2,3]. Another correct ordering is [0,2,1,3].

Example 3:

Input: numCourses = 1, prerequisites = [] **Output:** [0]

Constraints:

* `1 <= numCourses <= 2000` * `0 <= prerequisites.length <= numCourses * (numCourses - 1)` * `prerequisites[i].length == 2` * `0 <= ai, bi < numCourses` * `ai != bi` * All the pairs `[ai, bi]` are **distinct**.

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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