Course Schedule LeetCode

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] = [a_i, b_i] indicates that you **must** take course b_i first if you want to take course a_i.

- 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:

prerequisites = {{1, 0}}

The given Prerequisite courses:

To take course 1 you have to complete course 0 first.

Is it possible to finish all courses? Yes

Example 2:

prerequisites = {{1, 0}, {0, 1}}

To take course 1 you have to complete course 0 first.

To take course 0 you have to complete course 1 first.

Is it possible to finish all courses? No

Approach 1: Function to check if it's possible to finish all courses based on prerequisites

- **Explanation:**
    - The **canFinish** function constructs an adjacency list representing prerequisites and calculates in-degrees for each course.
    - It then performs a BFS traversal, updating in-degrees and enqueuing courses with in-degree 0.
    - The topological sort size is tracked during traversal.
    - If the topological sort size is equal to the number of courses, it is possible to finish all courses.
- **Time Complexity:**
    - **The time complexity is O(V + E), where V is the number of courses, and E is the number of prerequisites.**

        - **Both the creation of the adjacency list and BFS traversal contribute to the time complexity.**

- **Space Complexity:**
    - **The space complexity is O(V + E), where V is the number of courses, and E is the number of prerequisites.**
        - **The adjacency list, in-degrees vector, and the BFS queue contribute to space usage.**