LC 207. Course Schedule

Question

There are a total of *n* courses you have to take, labeled from `0` to `n-1`.

Some courses may have prerequisites, for example to take course 0 you have to first take course 1, which is expressed as a pair: `[0,1]`

Given the total number of courses and a list of prerequisite **pairs**, is it possible for you to finish all courses?

Example 1:

```
Input: 2, [[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: 2, [[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.
```

Solution

```python
class Solution:
    def canFinish(self, numCourses: int, prerequisites: List[List[int]]) -> bool:
        #Solution
        graph = collections.defaultdict(set)
        neighbors = collections.defaultdict(set)
        for course, prev in prerequisites:
            graph[course].add(prev)
            neighbors[prev].add(course)
        stack = [n for n in range(numCourses) if not graph[n]]
        count = 0
        while stack:
            node = stack.pop()
            count += 1
            for n in neighbors[node]:
                graph[n].remove(node)
                if not graph[n]:
                    stack.append(n)
        return count == numCourses
```