Find Champion II

solved by	Senan
Platform	LeetCode
difficulty	Medium
Serial	2924
tags	Topological Sort
language	C++
solved on	@26/11/2024
link	https://leetcode.com/problems/find-champion-ii/
Completion	☑

Intuition

The problem asks us to find a "champion" node in a directed graph, defined as the one with no incoming edges (in-degree 0). If there is more than one such node or none at all, the answer is -1. The idea is to keep track of the in-degree of each node and then verify if there is a unique node with in-degree 0.

Approach

1. **Tracking In-Degree**: Create a vector `inNodes` to store the in-degree for each node. Initialize it with zeros for all nodes.
2. **Update In-Degree**: Traverse through the edges, and for each directed edge from `edge[0]` to `edge[1]`, increment the in-degree of `edge[1]`.
3. **Find Champion**:
    - Traverse through the nodes and check their in-degrees.
    - If a node has in-degree 0, mark it as the potential "champion."
    - If more than one node is found with in-degree 0, return -1 as there cannot be more than one "champion."
4. **Edge Cases**: If no node has in-degree 0, return -1.

Complexity

Time Complexity:

- **O(E)**: To traverse all edges in the graph and update the in-degrees, where `E` is the number of edges.
- **O(N)**: To traverse through all nodes to find the "champion."
- **Total**: **O(N+E)**, where **N** is the number of nodes.

Space Complexity:

- **O(N)**: For the `inNodes` vector to store in-degrees for all nodes.
- **Auxiliary Space**: **O(1)**, as no additional structures are used.
- **Total**: **O(N)**

Code

```
class Solution {
public:
    int findChampion(int n, vector<vector<int>>& edges) {
        vector<int> inNodes(n);
        for (auto edge : edges) {
            inNodes[edge[1]]++;
        }

        int answer = INT_MIN;
        for (int i = 0; i < n; i++) {
            if (inNodes[i] == 0) {
                if (answer != INT_MIN) return -1;
                answer = i;
            }
        }
        return answer;
    }
};
```