Problem 1436: Destination City

Problem Information

Difficulty: Easy
Acceptance Rate: 79.45%
Paid Only: No
Tags: Array, Hash Table, String

Problem Description

You are given the array `paths`, where `paths[i] = [cityAi, cityBi]` means there exists a direct path going from `cityAi` to `cityBi`. _Return the destination city, that is, the city without any path outgoing to another city._

It is guaranteed that the graph of paths forms a line without any loop, therefore, there will be exactly one destination city.

Example 1:

Input: paths = [["London","New York"],["New York","Lima"],["Lima","Sao Paulo"]] **Output:** "Sao Paulo" **Explanation:** Starting at "London" city you will reach "Sao Paulo" city which is the destination city. Your trip consist of: "London" -> "New York" -> "Lima" -> "Sao Paulo".

Example 2:

Input: paths = [["B","C"],["D","B"],["C","A"]] **Output:** "A" **Explanation:** All possible trips are: "D" -> "B" -> "C" -> "A". "B" -> "C" -> "A". "C" -> "A". "A". Clearly the destination city is "A".

Example 3:

Input: paths = [["A","Z"]] **Output:** "Z"

Constraints:

* `1 <= paths.length <= 100` * `paths[i].length == 2` * `1 <= cityAi.length, cityBi.length <= 10` * `cityAi != cityBi` * All strings consist of lowercase and uppercase English letters and the space character.

Code Snippets

C++:

```
class Solution {
public:
string destCity(vector<vector<string>>& paths) {

}
};
```

Java:

```
class Solution {
public String destCity(List<List<String>> paths) {

}
}
```

Python3:

```
class Solution:
def destCity(self, paths: List[List[str]]) -> str:
```