Problem 3668: Restore Finishing Order

Problem Information

Difficulty: Easy
Acceptance Rate: 90.64%
Paid Only: No
Tags: Array, Hash Table

Problem Description

You are given an integer array `order` of length `n` and an integer array `friends`.

* `order` contains every integer from 1 to `n` **exactly once** , representing the IDs of the participants of a race in their **finishing** order. * `friends` contains the IDs of your friends in the race **sorted** in strictly increasing order. Each ID in friends is guaranteed to appear in the `order` array.

Return an array containing your friends' IDs in their **finishing** order.

Example 1:

Input: order = [3,1,2,5,4], friends = [1,3,4]

Output: [3,1,4]

Explanation:

The finishing order is `[_**3**_ , _**1**_ , 2, 5, _**4**_]`. Therefore, the finishing order of your friends is `[3, 1, 4]`.

Example 2:

Input: order = [1,4,5,3,2], friends = [2,5]

Output: [5,2]

Explanation:

The finishing order is `[1, 4, _**5**_ , 3, _**2**_]`. Therefore, the finishing order of your friends is `[5, 2]`.

Constraints:

* `1 <= n == order.length <= 100` * `order` contains every integer from 1 to `n` exactly once * `1 <= friends.length <= min(8, n)` * `1 <= friends[i] <= n` * `friends` is strictly increasing

Code Snippets

C++:

```
class Solution {
public:
vector<int> recoverOrder(vector<int>& order, vector<int>& friends) {

}
};
```

Java:

```
class Solution {
public int[] recoverOrder(int[] order, int[] friends) {

}
}
```

Python3:

```
class Solution:
def recoverOrder(self, order: List[int], friends: List[int]) -> List[int]:
```