

Description

Solution

Discuss (387)

Submissions

1762. Buildings With an Ocean View

Medium

👍 417

🔒 68

🤍 Add to List

🔗 Share

There are `n` buildings in a line. You are given an integer array `heights` of size `n` that represents the heights of the buildings in the line.

The ocean is to the right of the buildings. A building has an ocean view if the building can see the ocean without obstructions. Formally, a building has an ocean view if all the buildings to its right have a **smaller** height.

Return a list of indices (**0-indexed**) of buildings that have an ocean view, sorted in increasing order.

Example 1:

Input: heights = [4,2,3,1]

Output: [0,2,3]

Explanation: Building 1 (0-indexed) does not have an ocean view because building 2 is taller.

Example 2:

Input: heights = [4,3,2,1]

Output: [0,1,2,3]

Explanation: All the buildings have an ocean view.

Example 3:

Input: heights = [1,3,2,4]

Output: [3]

Explanation: Only building 3 has an ocean view.

Example 4:

Input: heights = [2,2,2,2]

Output: [3]

Explanation: Buildings cannot see the ocean if there are buildings of the **same** height to its right.

Constraints:

- `1 <= heights.length <= 105`
- `1 <= heights[i] <= 109`

Accepted 49,792

Submissions 61,231

Seen this question in a real interview before?

Yes

No

Companies 🏢 *i*

^

0 ~ 6 months

6 months ~ 1 year

1 year ~ 2 years

Facebook | 89

Microsoft | 2

Amazon | 2

Related Topics

^

Array

Stack

Monotonic Stack

Similar Questions

^

Number of Visible People in a Queue

Hard

Hide Hint 1

^

You can traverse the buildings from the nearest to the ocean to the furthest.

Hide Hint 2

^

Keep with you the maximum to the right while traversing to determine if you can see the ocean or not.

i Java

Autocomplete

```
1 class Solution {
2     public int[] findBuildings(int[] heights) {
3         int n = heights.length;
4         List<Integer> list = new ArrayList<>();
5         int maxHeight = -1;
6
7         for (int current = n - 1; current >= 0; --current) {
8             // If there is no building higher (or equal) than the current one to its right,
9             // push it in the list.
10            if (maxHeight < heights[current]) {
11                list.add(current);
12
13                // Update max building till now.
14                maxHeight = heights[current];
15            }
16        }
17
18        // Push building indices from list to answer array in reverse order.
19        int[] answer = new int[list.size()];
20        for (int i = 0; i < list.size(); ++i) {
21            answer[i] = list.get(list.size() - 1 - i);
22        }
23
24        return answer;
25    }
26 }
```

