33. Search in Rotated Sorted Array ★

Question | Editorial Solution

My Submissions (/problems/search-in-rotated-sorted-array/submissions/)

Total Accepted: **117163** Total Submissions: **377025** Difficulty: **Hard**

Notes

Suppose a sorted array is rotated at some pivot unknown to you beforehand.

(i.e., `0 1 2 4 5 6 7` might become `4 5 6 7 0 1 2`).

You are given a target value to search. If found in the array return its index, otherwise return -1.

You may assume no duplicate exists in the array.

Subscribe (/subscribe/) to see which companies asked this question

Show Tags

Show Similar Problems

Have you met this question in a real interview? Yes No

Discuss (https://leetcode.com/discuss/questions/oj/search-in-rotated-sorted-array)

Pick One (/problems/random-one-question/)

C++ </>

```
1  class Solution {
2  public:
3      int search(vector<int>& nums, int target) {
4          int first = 0, end = nums.size();
5
6          while (first != end) {
7              int mid = first + (end - first) / 2;
8              if (nums[mid] == target) {
9                  return mid;
10             }
11             else if (nums[mid] >= nums[first] && nums[mid] >= nums[end - 1]) { //
12                 if (target >= nums[first] && target < nums[mid]) {
13                     end = mid;
14                 }
15                 else {
16                     first = mid + 1;
17                 }
18             }
19             else if (nums[mid] <= nums[first] && nums[mid] <= nums[end - 1]) { //
20                 if (target > nums[mid] && target <= nums[end - 1]) {
21                     first = mid + 1;
22                 }
23                 else {
```

```
                    end = mid;
                }
            }
            else { // normal
                if (target > nums[mid]) { // right
                    first = mid + 1;
                }
                else { // left
                    end = mid;
                }
            }
        }
        return -1;
    }
};
```

Notes

Custom Testcase

Run Code | Submit Solution