

Search in Rotated Sorted Array

Submission Detail

196 / 196 test cases passed.

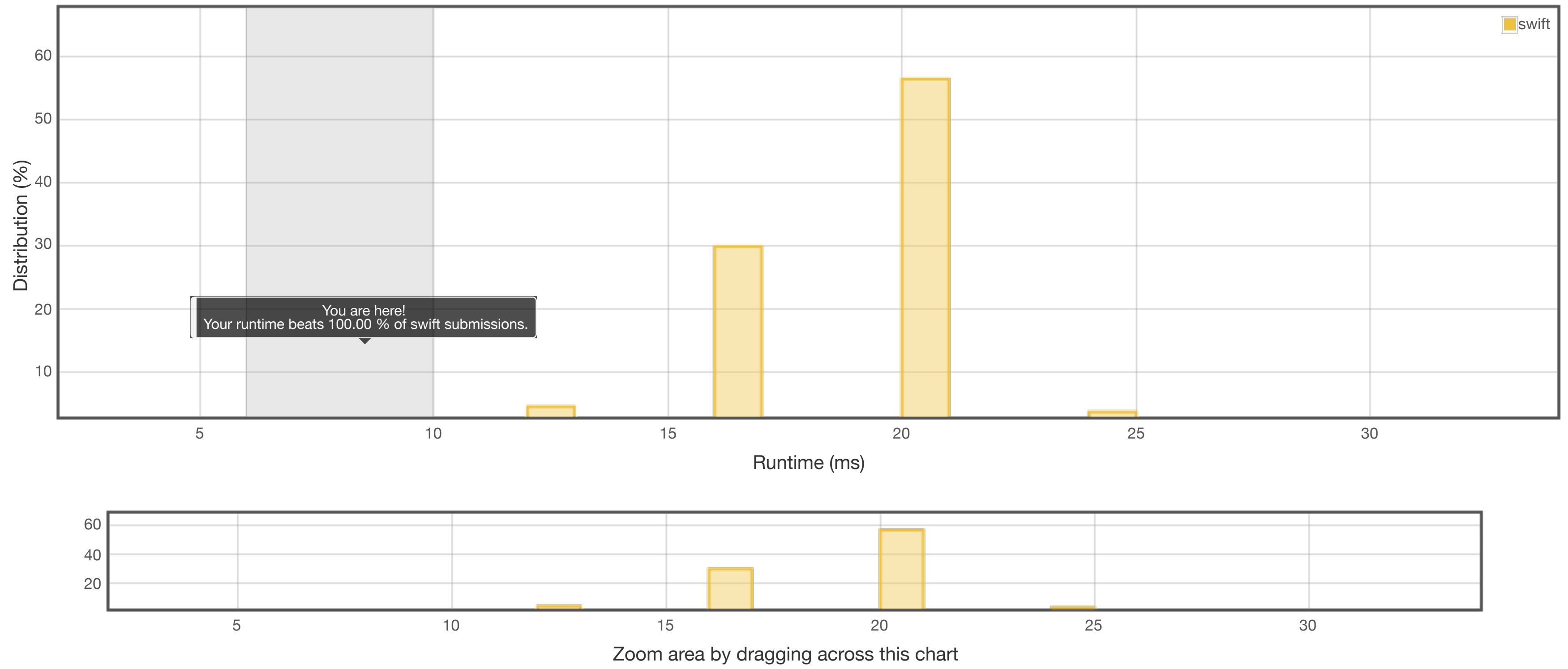
Runtime: 8 ms

Memory Usage: 21.2 MB

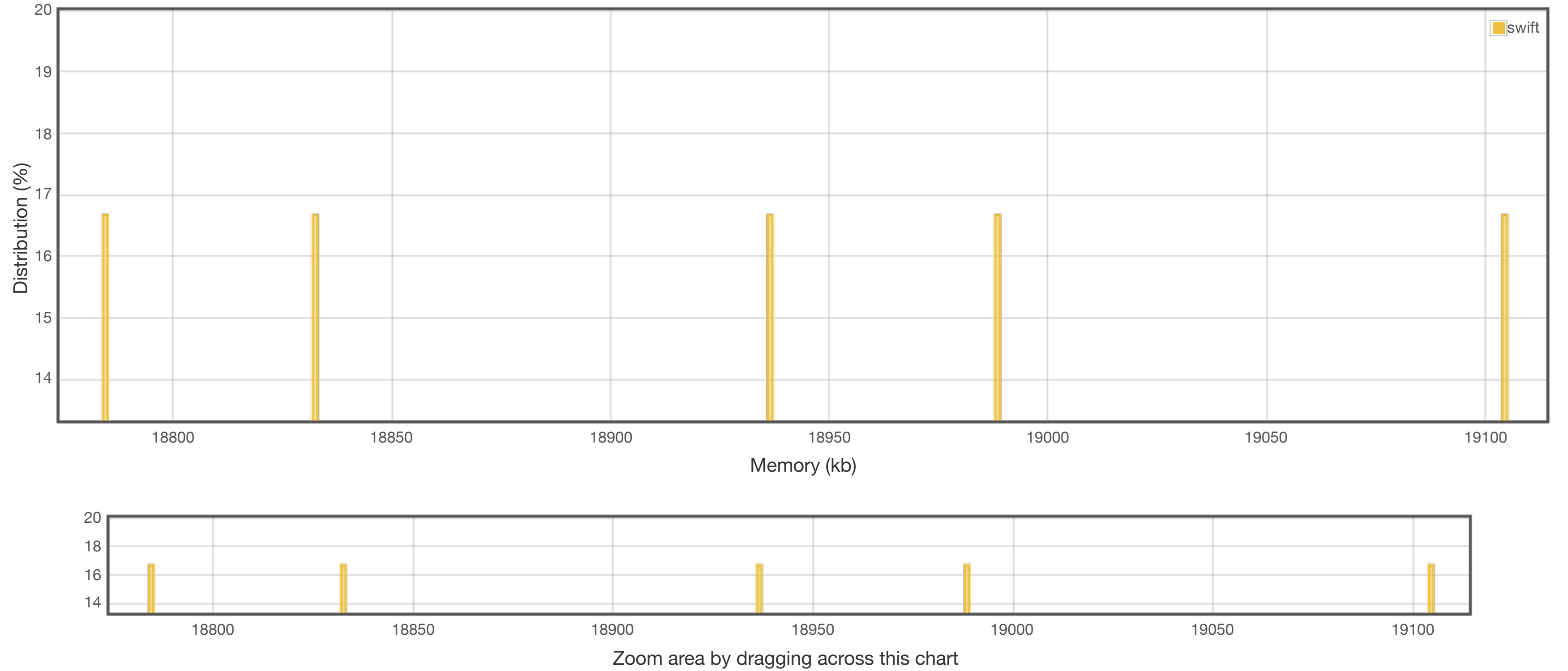
Status: Accepted

Submitted: 1 minute ago

Accepted Solutions Runtime Distribution



Accepted Solutions Memory Distribution



Invite friends to challenge Search in Rotated Sorted Array



Submitted Code: 1 minute ago

Language: swift

Edit Code

```
1 func myPrint(_ arg: Any) {
2     //print(arg)
3 }
4
5 class Solution {
6
7     func nonPivotedSearch(_ nums: [Int], _ l: Int, _ m: Int, _ h: Int, _ target: Int) -> Int {
8         myPrint("nonPivoted: l: \(l), m: \(m), h: \(h), target: \(target)")
9
10        if nums[m] == target { return m }
11        if m == h { // last 2 elements
12            return nums[l] == target ? l : -1
13        }
14        if (l == m) { // last 2 elements
15            return nums[h] == target ? h : -1
16        }
17
18        if target < nums[m] {
19            // go left
20            return nonPivotedSearch(nums, l, l + (m-l)/2, m, target)
21        } else {
22            // go right
23            return nonPivotedSearch(nums, m, m + (h-m)/2, h, target)
24        }
25    }
26
27    func pivotedSearch(_ nums: [Int], _ l: Int, _ m: Int, _ h: Int, _ target: Int) -> Int {
28        myPrint("Pivoted: l: \(l), m: \(m), h: \(h), target: \(target)")
29
30        // A. We found the target
31        if nums[m] == target { return m }
32
33        // B. Array with last 2 elements
34        if m == h {
35            return nums[l] == target ? l : -1
36        }
37        if (l == m) {
38            return nums[h] == target ? h : -1
39        }
40
41        // C. The pivot is m
42        if nums[m-1] > nums[m] {
43            if target > nums[m] && target <= nums[h] {
44                // go right
45                return nonPivotedSearch(nums, m, m + (h-m)/2, h, target)
46            } else {
47                // go left
48                return nonPivotedSearch(nums, l, l + (m-l)/2, m, target)
49            }
50        }
51
52        // D. The pivot is to the right
53        if nums[l] < nums[m] {
54            if target < nums[m] && target >= nums[l] {
55                // go left non-pivoted
56                return nonPivotedSearch(nums, l, l + (m-l)/2, m, target)
57            } else {
58                // go right pivoted
59                return pivotedSearch(nums, m, m + (h-m)/2, h, target)
60            }
61        }
62
63        // E. The pivot is to the left
64        if target > nums[m] && target <= nums[h] {
65            // go right non-pivoted
66            return nonPivotedSearch(nums, m, m + (h-m)/2, h, target)
67        } else {
68            // go left pivoted
69            return pivotedSearch(nums, l, l + (m-l)/2, m, target)
70        }
71    }
72
73    func search(_ nums: [Int], _ target: Int) -> Int {
74        guard nums.count > 0 else { return -1 }
75
76        let l = 0
77        let h = nums.count-1
78        let m = l + (h - l) / 2
79
80        return pivotedSearch(nums, l, m, h, target)
81    }
82 }
83
```

[Back to problem](#)