

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

Solution 2

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 // O(log(n)) time | O(1) space
4 function searchForRange(array, target) {
5   const finalRange = [-1, -1];
6   alteredBinarySearch(array, target, 0, array.length - 1, finalRange,
7   alteredBinarySearch(array, target, 0, array.length - 1, finalRange,
8   return finalRange;
9 }
10
11 function alteredBinarySearch(array, target, left, right, finalRange, )
12 while (left <= right) {
13   const mid = Math.floor((left + right) / 2);
14   if (array[mid] < target) {
15     left = mid + 1;
16   } else if (array[mid] > target) {
17     right = mid - 1;
18   } else {
19     if (goLeft) {
20       if (mid === 0 || array[mid - 1] !== target) {
21         finalRange[0] = mid;
22         return;
23       } else {
24         right = mid - 1;
25       }
26     } else {
27       if (mid === array.length - 1 || array[mid + 1] !== target) {
28         finalRange[1] = mid;
29         return;
30       } else {
31         left = mid + 1;
32       }
33     }
34   }
35 }
```

Solution 1

Solution 2

Solution 3

```
1 function searchForRange(array, target) {
2   // Write your code here.
3 }
4
5 // Do not edit the line below.
6 exports.searchForRange = searchForRange;
7
```

Our Tests

Custom Output

Submit Code

1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.

2

3 // O(log(n)) time | O(1) space

4 function searchForRange(array, target) {

5 const finalRange = [-1, -1];

6 alteredBinarySearch(array, target, 0, array.length - 1, finalRange,

7 alteredBinarySearch(array, target, 0, array.length - 1, finalRange,

8 return finalRange;

9 }

10

11 function alteredBinarySearch(array, target, left, right, finalRange, )

12 while (left <= right) {

13 const mid = Math.floor((left + right) / 2);

14 if (array[mid] < target) {

15 left = mid + 1;

16 } else if (array[mid] > target) {

17 right = mid - 1;

18 } else {

19 if (goLeft) {

20 if (mid === 0 || array[mid - 1] !== target) {

21 finalRange[0] = mid;

22 return;

23 } else {

24 right = mid - 1;

25 }

26 } else {

27 if (mid === array.length - 1 || array[mid + 1] !== target) {

28 finalRange[1] = mid;

29 return;

30 } else {

31 left = mid + 1;

32 }

33 }

34 }

35 }

1 function searchForRange(array, target) {

2 // Write your code here.

3 }

4

5 // Do not edit the line below.

6 exports.searchForRange = searchForRange;

7

```
10 # Run Test Case 401 - Success (1/1)
11 # Run expectProgramOutputFileRange(25, 1, 1, 8, 8, 245, 71) on test
12 # file
13
14 # Run Test Case 402 - Success (1/1)
15 # Run expectProgramOutputFileRange(25, 1, 1, 8, 8, 245, 81) on test
16 # file
17
18 # Run Test Case 403 - Success (1/1)
19 # Run expectProgramOutputFileRange(25, 1, 1, 8, 8, 245, 245) on test
20 # file
21
22 # Run Test Case 404 - Success (1/1)
23 # Run expectProgramOutputFileRange(25, 1, 1, 8, 8, 245, 71) on test
24 # file
```

Run or submit code when you're ready.