

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

Solution 2

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 #include <vector>
4 using namespace std;
5
6 int binarySearch(vector<int> array, int target);
7 int binarySearchHelper(vector<int> array, int target, int left, int right) {
8
9     // O(log(n)) time | O(1) space
10    int binarySearch(vector<int> array, int target) {
11        return binarySearchHelper(array, target, 0, array.size() - 1);
12    }
13
14    int binarySearchHelper(vector<int> array, int target, int left, int right) {
15        if (left > right) {
16            return -1;
17        }
18        int middle = (left + right) / 2;
19        int potentialMatch = array[middle];
20        if (target == potentialMatch) {
21            return middle;
22        } else if (target < potentialMatch) {
23            return binarySearchHelper(array, target, left, middle - 1);
24        } else {
25            return binarySearchHelper(array, target, middle + 1, right);
26        }
27    }
28 }
```

Solution 1

Solution 2

Solution 3

```
1 #include <vector>
2 using namespace std;
3
4 int binarySearch(vector<int> array, int target) {
5     // Write your code here.
6     return -1;
7 }
8
```

Our Tests

Custom Output

Submit Code

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

2

3 #include <vector>

4 using namespace std;

5

6 int binarySearch(vector<int> array, int target);

7 int binarySearchHelper(vector<int> array, int target, int left, int right) {

8

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

10 int binarySearch(vector<int> array, int target) {

11 return binarySearchHelper(array, target, 0, array.size() - 1);

12 }

13

14 int binarySearchHelper(vector<int> array, int target, int left, int right) {

15 if (left > right) {

16 return -1;

17 }

18 int middle = (left + right) / 2;

19 int potentialMatch = array[middle];

20 if (target == potentialMatch) {

21 return middle;

22 } else if (target < potentialMatch) {

23 return binarySearchHelper(array, target, left, middle - 1);

24 } else {

25 return binarySearchHelper(array, target, middle + 1, right);

26 }

27 }

28 }

1 #include <vector>

2 using namespace std;

3

4 int binarySearch(vector<int> array, int target) {

5 // Write your code here.

6 return -1;

7 }

8

```
18 def test_test_case_27():
19     assert test_case_27(25, 5, 25, 2500, 50) == 50
20
21 def test_test_case_27():
22     assert test_case_27(25, 5, 25, 2500, 50) == 50
23
24 def test_test_case_27():
25     assert test_case_27(25, 5, 25, 25, 25, 25, 25, 25, 25, 25)
26
27 def test_test_case_27():
28     assert test_case_27(25, 5, 25, 25, 25, 25, 25, 25, 25, 25)
```

Run or submit code when you're ready.