

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 shiftedBinarySearch(vector<int> array, int target);
7 int shiftedBinarySearchHelper(vector<int> array, int target, int left
8                               int right);
9
10 // O(log(n)) time | O(log(n)) space
11 int shiftedBinarySearch(vector<int> array, int target) {
12     return shiftedBinarySearchHelper(array, target, 0, array.size() - 1
13 }
14
15 int shiftedBinarySearchHelper(vector<int> array, int target, int left,
16                               int right) {
17     if (left > right) {
18         return -1;
19     }
20     int middle = (left + right) / 2;
21     int potentialMatch = array[middle];
22     int leftNum = array[left];
23     int rightNum = array[right];
24     if (target == potentialMatch) {
25         return middle;
26     } else if (leftNum <= potentialMatch) {
27         if (target < potentialMatch && target >= leftNum) {
28             return shiftedBinarySearchHelper(array, target, left, middle - 1
29         } else {
30             return shiftedBinarySearchHelper(array, target, middle + 1, righ
31         }
32     } else {
33         if (target > potentialMatch && target <= rightNum) {
```

Solution 1

Solution 2

Solution 3

```
1 #include <vector>
2 using namespace std;
3
4 int shiftedBinarySearch(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 shiftedBinarySearch(vector<int> array, int target);
7 int shiftedBinarySearchHelper(vector<int> array, int target, int left
8                               int right);
9
10 // O(log(n)) time | O(log(n)) space
11 int shiftedBinarySearch(vector<int> array, int target) {
12     return shiftedBinarySearchHelper(array, target, 0, array.size() - 1
13 }
14
15 int shiftedBinarySearchHelper(vector<int> array, int target, int left,
16                               int right) {
17     if (left > right) {
18         return -1;
19     }
20     int middle = (left + right) / 2;
21     int potentialMatch = array[middle];
22     int leftNum = array[left];
23     int rightNum = array[right];
24     if (target == potentialMatch) {
25         return middle;
26     } else if (leftNum <= potentialMatch) {
27         if (target < potentialMatch && target >= leftNum) {
28             return shiftedBinarySearchHelper(array, target, left, middle - 1
29         } else {
30             return shiftedBinarySearchHelper(array, target, middle + 1, righ
31         }
32     } else {
33         if (target > potentialMatch && target <= rightNum) {
```

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 #include <vector>
4 using namespace std;
5
6 int shiftedBinarySearch(vector<int> array, int target);
7 int shiftedBinarySearchHelper(vector<int> array, int target, int left
8                               int right);
9
10 // O(log(n)) time | O(log(n)) space
11 int shiftedBinarySearch(vector<int> array, int target) {
12     return shiftedBinarySearchHelper(array, target, 0, array.size() - 1
13 }
14
15 int shiftedBinarySearchHelper(vector<int> array, int target, int left,
16                               int right) {
17     if (left > right) {
18         return -1;
19     }
20     int middle = (left + right) / 2;
21     int potentialMatch = array[middle];
22     int leftNum = array[left];
23     int rightNum = array[right];
24     if (target == potentialMatch) {
25         return middle;
26     } else if (leftNum <= potentialMatch) {
27         if (target < potentialMatch && target >= leftNum) {
28             return shiftedBinarySearchHelper(array, target, left, middle - 1
29         } else {
30             return shiftedBinarySearchHelper(array, target, middle + 1, righ
31         }
32     } else {
33         if (target > potentialMatch && target <= rightNum) {
```

```
1 def test():
2     assert testFunction(100, 5, 5, 100, 5) == 100
3
4 def test():
5     assert testFunction(100, 100, 5, 5, 100) == 100
6
7 def test():
8     assert testFunction(100, 100, 10, 10, 10, 5, 10, 10, 10)
9     == 100
10
11 def test():
12     assert testFunction(100, 10, 10, 10, 10, 5, 10, 10, 10)
13     == 100
```

Run or submit code when you're ready.