

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(1) 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     while (left <= right) {
18         int middle = (left + right) / 2;
19         int potentialMatch = array[middle];
20         int leftNum = array[left];
21         int rightNum = array[right];
22         if (target == potentialMatch) {
23             return middle;
24         } else if (leftNum <= potentialMatch) {
25             if (target < potentialMatch && target >= leftNum) {
26                 right = middle - 1;
27             } else {
28                 left = middle + 1;
29             }
30         } else {
31             if (target > potentialMatch && target <= rightNum) {
32                 left = middle + 1;
33             } else {
34                 right = middle - 1;
35             }
36         }
37     }
38     return -1;
39 }
```

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(1) 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 while (left <= right) {

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

19 int potentialMatch = array[middle];

20 int leftNum = array[left];

21 int rightNum = array[right];

22 if (target == potentialMatch) {

23 return middle;

24 } else if (leftNum <= potentialMatch) {

25 if (target < potentialMatch && target >= leftNum) {

26 right = middle - 1;

27 } else {

28 left = middle + 1;

29 }

30 } else {

31 if (target > potentialMatch && target <= rightNum) {

32 left = middle + 1;

33 } else {

34 right = middle - 1;

35 }

36 }

37 }

38 return -1;

39 }

Custom Output

Submit Code

```
1 def sum(*args):
2     return sum(args)
3
4 # Test cases
5 sum(1, 2, 3, 4, 5)  # 15
6 sum(10, 20, 30)    # 60
7 sum(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)  # 55
8 sum(100, 200, 300)  # 600
9 sum(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15)  # 120
10 sum(1000, 2000, 3000)  # 6000
11 sum(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20)  # 190
```

Run or submit code when you're ready.