

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 vector<int> searchForRange(vector<int> array, int target);
7 void alteredBinarySearch(vector<int> array, int target, int left, int
8                           vector<int> *finalRange, bool goLeft);
9
10 // O(log(n)) time | O(1) space
11 vector<int> searchForRange(vector<int> array, int target) {
12     vector<int> finalRange{-1, -1};
13     alteredBinarySearch(array, target, 0, array.size() - 1, &finalRange,
14     alteredBinarySearch(array, target, 0, array.size() - 1, &finalRange,
15     return finalRange;
16 }
17
18 void alteredBinarySearch(vector<int> array, int target, int left, int
19                           vector<int> *finalRange, bool goLeft) {
20     while (left <= right) {
21         int mid = (left + right) / 2;
22         if (array[mid] < target) {
23             left = mid + 1;
24         } else if (array[mid] > target) {
25             right = mid - 1;
26         } else {
27             if (goLeft) {
28                 if (mid == 0 || array[mid - 1] != target) {
29                     finalRange->at(0) = mid;
30                     return;
31                 } else {
32                     right = mid - 1;
33                 }

```

Solution 1

Solution 2

Solution 3

```
1 #include <vector>
2 using namespace std;
3
4 vector<int> searchForRange(vector<int> array, int target) {
5     // Write your code here.
6     return {};
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 vector<int> searchForRange(vector<int> array, int target);
7 void alteredBinarySearch(vector<int> array, int target, int left, int
8                           vector<int> *finalRange, bool goLeft);
9
10 // O(log(n)) time | O(1) space
11 vector<int> searchForRange(vector<int> array, int target) {
12     vector<int> finalRange{-1, -1};
13     alteredBinarySearch(array, target, 0, array.size() - 1, &finalRange,
14     alteredBinarySearch(array, target, 0, array.size() - 1, &finalRange,
15     return finalRange;
16 }
17
18 void alteredBinarySearch(vector<int> array, int target, int left, int
19                           vector<int> *finalRange, bool goLeft) {
20     while (left <= right) {
21         int mid = (left + right) / 2;
22         if (array[mid] < target) {
23             left = mid + 1;
24         } else if (array[mid] > target) {
25             right = mid - 1;
26         } else {
27             if (goLeft) {
28                 if (mid == 0 || array[mid - 1] != target) {
29                     finalRange->at(0) = mid;
30                     return;
31                 } else {
32                     right = mid - 1;
33                 }

```

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

```

