Description | Accepted | Editorial | Solutions | Submissions

← All Submissions

Accepted 114 / 114 testcases passed

amit_89333 submitted at Feb 10, 2026 17:01

Editorial | Solution

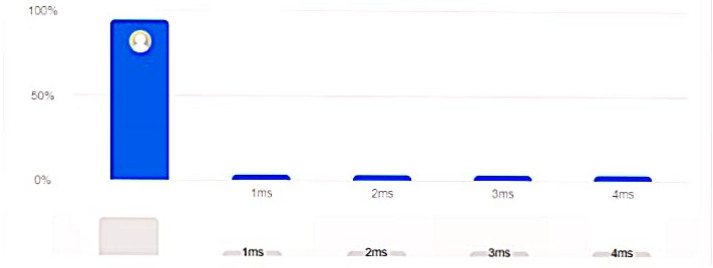
Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

43.19 MB Beats 93.71%



Code | Java

```java
class Solution {
    public int[] plusOne(int[] digits) {

        // Start from last digit
        for (int i = digits.length - 1; i >= 0; i--) {
```

Code

Java | Auto

```java
        }

        // If all digits were 9
        int[] result = new int[digits.length + 1];
        result[0] = 1;

        return result;
    }
}
```

Saved

Ln 23, Col 1

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 | Case 2 | Case 3

Input

digits =

[1,2,3]

Output

[1,2,4]

Expected

Description | Accepted × | Editorial | Solutions | Submissions

← All Submissions

Accepted 128 / 128 testcases passed

amit_89333 submitted at Feb 10, 2026 16:51

Editorial | Solution

Runtime

7 ms | Beats 76.55%

Analyze Complexity

Memory

48.26 MB | Beats 99.94%

Code | Java

```java
import java.util.*;

class Solution {
    public List<List<String>> groupAnagrams(String[] strs) {
```

Code

Java | Auto

```java
                map.put(key, new ArrayList<>());
            }
            map.get(key).add(s);
        }

        // Step 3: Return all grouped values
        return new ArrayList<>(map.values());
    }
}
```

Saved

Ln 26, C

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 | Case 2 | Case 3

Input

strs =

```
["eat","tea","tan","ate","nat","bat"]
```

Output

```
[["eat","tea","ate"],["bat"],["tan","nat"]]
```

Expected

← All Submissions

Accepted 110 / 110 testcases passed

amit_89333 submitted at Feb 10, 2026 16:48

Editorial | Solution

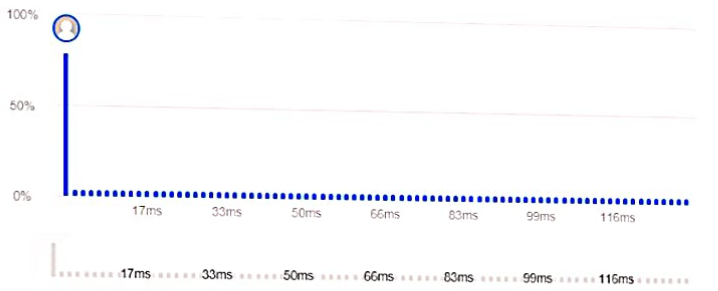
Runtime

1 ms | Beats 99.53%

Analyze Complexity

Memory

47.61 MB Beats 6.56%



Code | Java

```java
class Solution {
    public int jump(int[] nums) {

        int jumps = 0;
        int currentEnd = 0;
```

Code

Java | Auto

```java
class Solution {
    public int jump(int[] nums) {

        int jumps = 0;
        int currentEnd = 0;
        int farthest = 0;

        for (int i = 0; i < nums.length - 1; i++) {

            farthest = Math.max(farthest, i + nums[i]);
```

Saved

Ln 22, Col 1

Testcase | Test Result

You must run your code first

← All Submissions

Accepted 160 / 160 testcases passed

amit_89333 submitted at Feb 10, 2026 16:43

Editorial | Solution

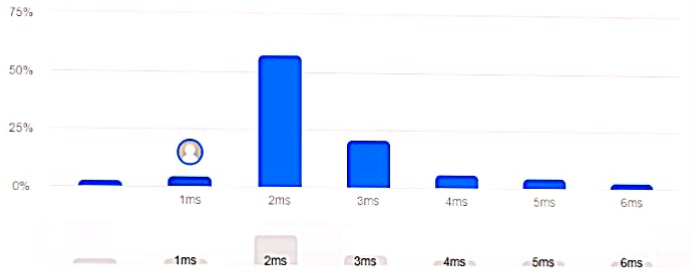
Runtime

1 ms | Beats 99.78%

Analyze Complexity

Memory

46.00 MB Beats 19.64%



Code | Java

```java
class Solution {
    public List<List<Integer>> combinationSum(int[] candidates, int target)
        List<List<Integer>> ans = new ArrayList<>();
        List<Integer> path = new ArrayList<>();
        soln(candidates,target,0,path,ans);
```

</> Code

Java | Auto

```java
            soln(arr,target-arr[j],j,path,ans);
            path.remove(path.size() - 1);
        }

    }
}
```

Saved

Ln 23, Col 2

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 | Case 2 | Case 3

Input

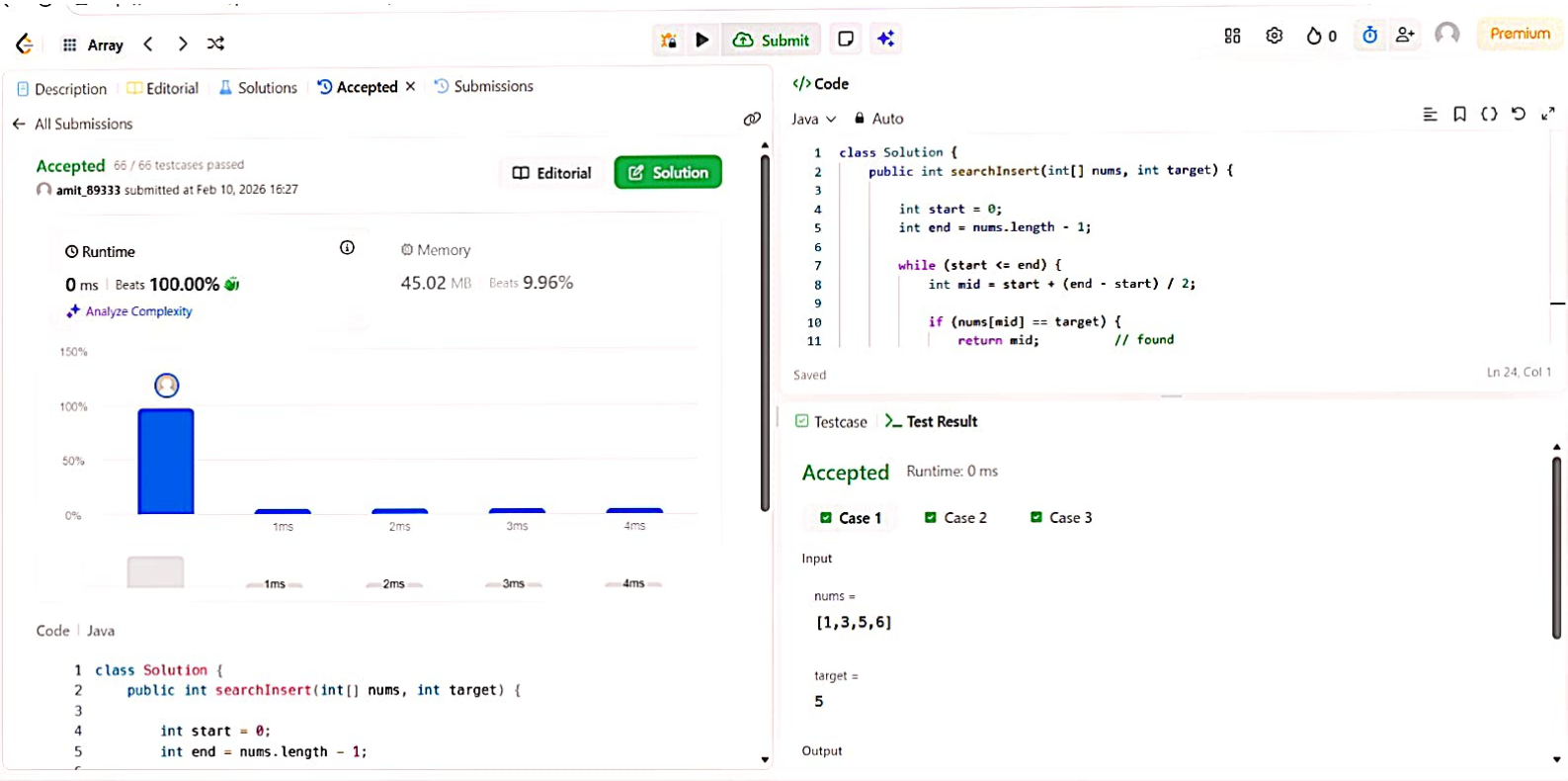
candidates =

[2,3,6,7]

target =

7

Output


Array
Submit
Premium
Description
Editorial
Solutions
Accepted
Submissions
All Submissions
Accepted 66 / 66 testcases passed
amit_89333 submitted at Feb 10, 2026 16:27
Editorial
Solution
Runtime
0 ms | Beats 100.00%
Analyze Complexity
Memory
45.02 MB | Beats 9.96%
150%
100%
50%
0%
1ms
2ms
3ms
4ms
Code | Java
1 class Solution {
2 public int searchInsert(int[] nums, int target) {
3
4 int start = 0;
5 int end = nums.length - 1;
Code
Java
Auto
1 class Solution {
2 public int searchInsert(int[] nums, int target) {
3
4 int start = 0;
5 int end = nums.length - 1;
6
7 while (start <= end) {
8 int mid = start + (end - start) / 2;
9
10 if (nums[mid] == target) {
11 return mid; // found
Saved
Ln 24, Col 1
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Case 3
Input
nums =
[1,3,5,6]
target =
5
Output

Description | Editorial | Solutions | Accepted ✕ | Submissions

← All Submissions

Accepted 176 / 176 testcases passed

amit_89333 submitted at Feb 10, 2026 16:45

Editorial | Solution

Runtime

5 ms | Beats 99.22%

Analyze Complexity

Memory

44.84 MB | Beats 96.75%

Code | Java

```java
class Solution {
    public List<List<Integer>> combinationSum2(int[] candidates, int target)
        Arrays.sort(candidates);
        List<List<Integer>> ans = new ArrayList<>();
        List<Integer> path = new ArrayList<>();
```

Code

Java | Auto

```java
            if(j>i && arr[j]==arr[j-1] ) continue;
            if (arr[j] > target) break;
            path.add(arr[j]);
            soln(arr,target-arr[j],j+1,path,ans);
            path.remove(path.size() - 1);
        }

    }
}
```

Saved | Ln 25, Col 1

Testcase | Test Result

Accepted Runtime: 1 ms

Case 1 | Case 2

Input

candidates =

[10,1,2,7,6,1,5]

target =

8

Output

Description | Accepted × | Editorial | Solutions | Submissions

← All Submissions

Accepted 114 / 114 testcases passed

amit_89333 submitted at Feb 10, 2026 17:01

Editorial | Solution

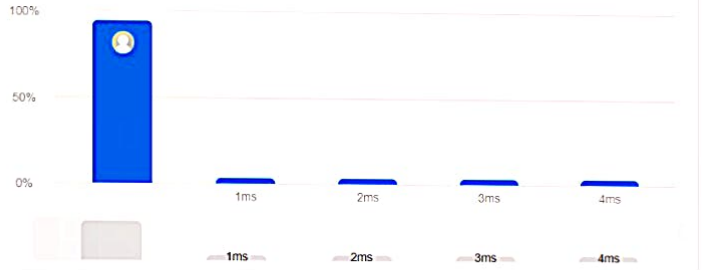
Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

43.19 MB | Beats 93.71%



Code | Java

```java
class Solution {
    public int[] plusOne(int[] digits) {

        // Start from last digit
        for (int i = digits.length - 1; i >= 0; i--) {
```

Code

Java | Auto

```java
        }

        // If all digits were 9
        int[] result = new int[digits.length + 1];
        result[0] = 1;

        return result;
    }
}
```

Saved

Ln 23, Col 1

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 | Case 2 | Case 3

Input

digits =

[1,2,3]

Output

[1,2,4]

Expected

← All Submissions

Accepted 110 / 110 testcases passed

amit_89333 submitted at Feb 10, 2026 16:48

Editorial | Solution

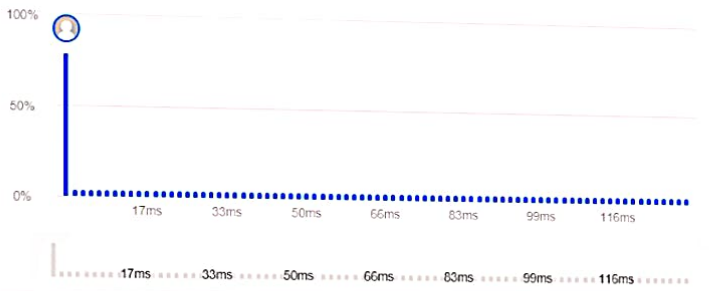
Runtime

1 ms | Beats 99.53%

Analyze Complexity

Memory

47.61 MB Beats 6.56%



Code | Java

```java
class Solution {
    public int jump(int[] nums) {

        int jumps = 0;
        int currentEnd = 0;
```

Code

Java | Auto

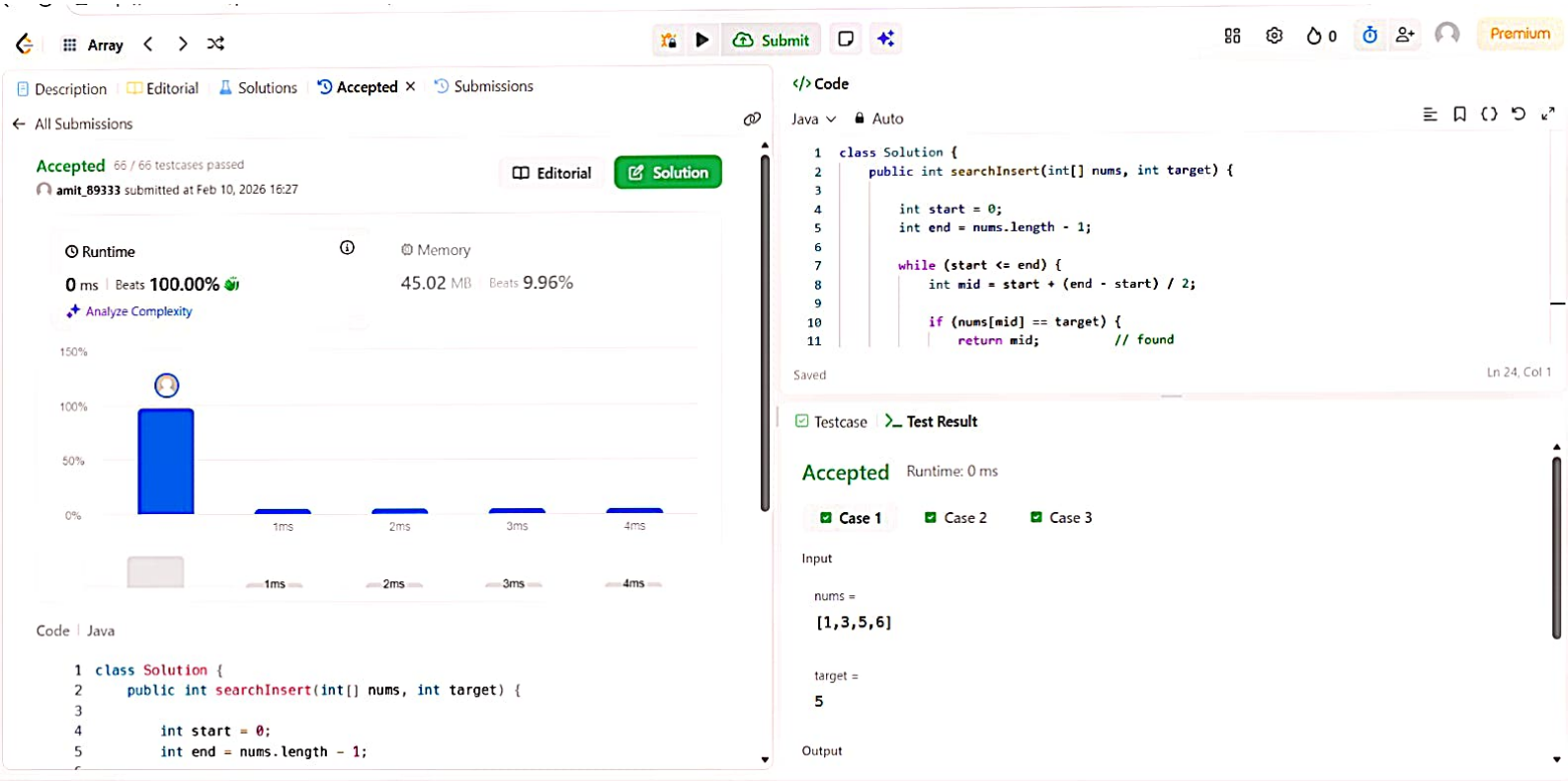
```java
class Solution {
    public int jump(int[] nums) {

        int jumps = 0;
        int currentEnd = 0;
        int farthest = 0;

        for (int i = 0; i < nums.length - 1; i++) {

            farthest = Math.max(farthest, i + nums[i]);
```

Saved

Ln 22, Col 1

Testcase | Test Result

You must run your code first


Array
Submit
Premium
Description
Editorial
Solutions
Accepted
Submissions
All Submissions
Accepted 66 / 66 testcases passed
amit_89333 submitted at Feb 10, 2026 16:27
Editorial
Solution
Runtime
0 ms | Beats 100.00%
Analyze Complexity
Memory
45.02 MB | Beats 9.96%
150%
100%
50%
0%
1ms
2ms
3ms
4ms
Code | Java
1 class Solution {
2 public int searchInsert(int[] nums, int target) {
3
4 int start = 0;
5 int end = nums.length - 1;
Code
Java
Auto
1 class Solution {
2 public int searchInsert(int[] nums, int target) {
3
4 int start = 0;
5 int end = nums.length - 1;
6
7 while (start <= end) {
8 int mid = start + (end - start) / 2;
9
10 if (nums[mid] == target) {
11 return mid; // found
Saved
Ln 24, Col 1
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Case 3
Input
nums =
[1,3,5,6]
target =
5
Output

Description | Editorial | Solutions | Accepted ✕ | Submissions

← All Submissions

Accepted 176 / 176 testcases passed

amit_89333 submitted at Feb 10, 2026 16:45

Editorial | Solution

Runtime

5 ms | Beats 99.22%

Analyze Complexity

Memory

44.84 MB | Beats 96.75%

Code | Java

```java
class Solution {
    public List<List<Integer>> combinationSum2(int[] candidates, int target)
        Arrays.sort(candidates);
        List<List<Integer>> ans = new ArrayList<>();
        List<Integer> path = new ArrayList<>();
```

Code

Java | Auto

```java
            if(j>i && arr[j]==arr[j-1] ) continue;
            if (arr[j] > target) break;
            path.add(arr[j]);
            soln(arr,target-arr[j],j+1,path,ans);
            path.remove(path.size() - 1);
        }

    }
}
```

Saved | Ln 25, Col 1

Testcase | Test Result

Accepted Runtime: 1 ms

Case 1 | Case 2

Input

candidates =

[10,1,2,7,6,1,5]

target =

8

Output

Description | Accepted ✕ | Editorial | Solutions | Submissions

← All Submissions

Accepted 128 / 128 testcases passed

amit_89333 submitted at Feb 10, 2026 16:51

Editorial | Solution

Runtime

7 ms | Beats 76.55%

Analyze Complexity

Memory

48.26 MB | Beats 99.94%

Code | Java

```java
import java.util.*;

class Solution {
    public List<List<String>> groupAnagrams(String[] strs) {
```

Java | Auto

```java
                map.put(key, new ArrayList<>());
            }
            map.get(key).add(s);
        }

        // Step 3: Return all grouped values
        return new ArrayList<>(map.values());
    }
}
```

Saved

Ln 26, C

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 | Case 2 | Case 3

Input

strs =

```
["eat","tea","tan","ate","nat","bat"]
```

Output

```
[["eat","tea","ate"],["bat"],["tan","nat"]]
```

Expected

← All Submissions

Accepted 160 / 160 testcases passed

amit_89333 submitted at Feb 10, 2026 16:43

Editorial | Solution

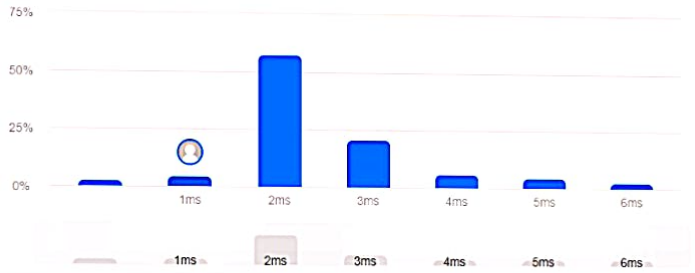
Runtime

1 ms | Beats 99.78%

Analyze Complexity

Memory

46.00 MB Beats 19.64%



Code | Java

```java
class Solution {
    public List<List<Integer>> combinationSum(int[] candidates, int target)
        List<List<Integer>> ans = new ArrayList<>();
        List<Integer> path = new ArrayList<>();
        soln(candidates,target,0,path,ans);
```

</> Code

Java | Auto

```java
            soln(arr,target-arr[j],j,path,ans);
            path.remove(path.size() - 1);
        }

    }
}
```

Saved

Ln 23, Col 2

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 | Case 2 | Case 3

Input

candidates =

[2,3,6,7]

target =

7

Output

Code

Java | Auto

```java
class Solution {
    public int[] searchRange(int[] nums, int target) {
        int[] ans = {-1, -1};

        // First occurrence
        ans[0] = binarySearch(nums, target, true);
        // Last occurrence
        ans[1] = binarySearch(nums, target, false);

        return ans;
    }

    // Helper method to find first or last position
```

Saved | Ln 1, Col 1

Description | Editorial | Solutions | Submissions

	Status	Language	Runtime	Memory	Notes
2	Accepted a few seconds ago	Java	0 ms	48.5 MB	
1	Accepted Feb 13, 2026	Java	0 ms	48.1 MB	

Testcase | Test Result | Accepted ×

All Submissions

Accepted 88 / 88 testcases passed

amit_89333 submitted at Feb 15, 2026 23:38

Editorial | Solution

Runtime | Memory

0 ms | Beats 100.00% | 48.54 MB | Beats 11.72%

Analyze Complexity

Code | Java

```java
class Solution {
    public int[] searchRange(int[] nums, int target) {
        int[] ans = {-1, -1};

        // First occurrence
```

Java | Auto

```java
class Solution {
    public int search(int[] nums, int target) {
        int start = 0, end = nums.length - 1;

        while (start <= end) {
            int mid = start + (end - start) / 2;

            if (nums[mid] == target) return mid;

            // Check if left part is sorted
            if (nums[mid] >= nums[start]) {
                if (target >= nums[start] && target < nums[mid]) {
                    end = mid - 1; // move left
```

Saved | Ln 1, Col 1

Description | Editorial | Solutions | Submissions

	Status	Language	Runtime	Memory	Notes
1	Accepted Feb 13, 2026	Java	0 ms	44 MB	

← All Submissions

Accepted 196 / 196 testcases passed

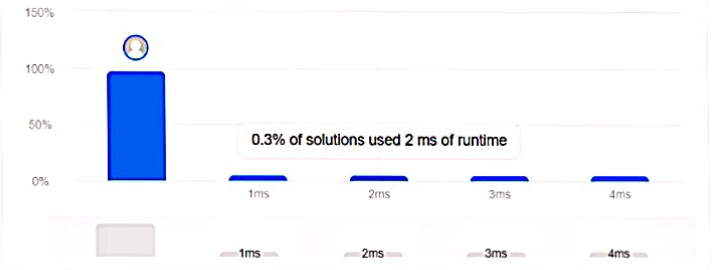
amit_89333 submitted at Feb 13, 2026 21:17

Editorial | Solution

Runtime | Memory

0 ms | Beats 100.00% | 43.97 MB | Beats 44.98%

Analyze Complexity



Code | Java

```java
class Solution {
    public int search(int[] nums, int target) {
        int start = 0, end = nums.length - 1;

        while (start <= end) {
```