



Master algorithms together on [Binary Search](#)! Create a room, invite your friends, and race to finish the problems.

Daily Coding Problem

[Blog](#)

Daily Coding Problem #257

Problem

This problem was asked by WhatsApp.

Given an array of integers out of order, determine the bounds of the smallest window that must be sorted in order for the entire array to be sorted. For example, given [3, 7, 5, 6, 9], you should return (1, 3).

Solution

One method we can try is to first find out what the array elements would look like when sorted. For example, [3, 7, 5, 6, 9], after sorting, becomes [3, 5, 6, 7, 9]. We can see that the first and last elements remain unchanged, whereas the middle elements are altered. Therefore, it suffices to take the first and last altered elements as our window.

```
def window(array):  
    left, right = None, None  
    s = sorted(array)  
  
    for i in range(len(array)):  
        if array[i] != s[i] and left is None:  
            left = i  
  
        elif array[i] != s[i]:  
            right = i
```

```
        right = i

    return left, right
```

This solution takes $O(N \log N)$ time and space, since we create a sorted copy of the original array.

Suppose instead that we traversed the array, from left to right, and took note of whether each element was less than the maximum seen up to that point. This element would have to be part of the sorting window, since we would have to move the maximum element past it.

As a result, we can take the last element that is less than the running maximum, and use it as our right bound. Similarly, for our left bound, we can traverse the array from right to left, and find the last element that exceeds the running minimum.

This will take two passes over the array, operating in $O(N)$ time and $O(1)$ space.

```
def window(array):
    left, right = None, None
    n = len(array)
    max_seen, min_seen = -float("inf"), float("inf")

    for i in range(n):
        max_seen = max(max_seen, array[i])
        if array[i] < max_seen:
            right = i

    for i in range(n - 1, -1, -1):
        min_seen = min(min_seen, array[i])
        if array[i] > min_seen:
            left = i

    return left, right
```

© Daily Coding Problem 2019

[Privacy Policy](#)

[Terms of Service](#)

[Press](#)