

167. Two Sum II - Input array is sorted

🕒 Created	@June 26, 2021 5:34 AM
▼ Difficulty	Medium
☰ LC Url	https://leetcode.com/problems/two-sum-ii-input-array-is-sorted/
▼ Importance	
☰ Tag	Two pointers
☰ Video	

Given an array of integers `numbers` that is already **sorted in non-decreasing order**, find two numbers such that they add up to a specific `target` number.

Return *the indices of the two numbers (1-indexed)* as an integer array `answer` of size `2`, where `1 <= answer[0] < answer[1] <= numbers.length`.

The tests are generated such that there is **exactly one solution**. You **may not** use the same element twice.

Example 1:

```
Input: numbers = [2,7,11,15], target = 9
Output: [1,2]
Explanation: The sum of 2 and 7 is 9. Therefore index1 = 1, index2 = 2.
```

Example 2:

```
Input: numbers = [2,3,4], target = 6
Output: [1,3]
```

Example 3:

Input: numbers = [-1,0], target = -1
Output: [1,2]

Constraints:

- `2 <= numbers.length <= 3 * 104`
- `1000 <= numbers[i] <= 1000`
- `numbers` is sorted in **non-decreasing order**.
- `1000 <= target <= 1000`
- The tests are generated such that there is **exactly one solution**.

Solution

```
class Solution:
    def twoSum(self, numbers: List[int], target: int) -> List[int]:
        start = 0
        end = len(numbers) - 1

        while start != end:
            if numbers[start] + numbers[end] > target:
                end -= 1
            elif numbers[start] + numbers[end] < target:
                start += 1
            else:
                return [start+1, end+1]
```

```
class Solution:
    def twoSum(self, numbers: List[int], target: int) -> List[int]:
        left, right = 0, len(numbers) - 1

        while left < right:
            isum = numbers[left] + numbers[right]

            if isum == target:
                return [left+1, right+1]
            elif isum < target:
                left += 1
            else:
                right -= 1
        return [-1, -1]
```

