Problem 167: Two Sum II - Input Array Is Sorted

Problem Information

Difficulty: Medium
Acceptance Rate: 64.13%
Paid Only: No
Tags: Array, Two Pointers, Binary Search

Problem Description

Given a **1-indexed** 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. Let these two numbers be `numbers[index1]` and `numbers[index2]` where `1 <= index1 < index2 <= numbers.length`.

Return _the indices of the two numbers,_`index1` _and_`index2` _,**added by one** as an integer array _`[index1, index2]`_of length 2._

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

Your solution must use only constant extra space.

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. We return [1, 2].

Example 2:

Input: numbers = [_2_ ,3,_4_], target = 6 **Output:** [1,3] **Explanation:** The sum of 2 and 4 is 6. Therefore index1 = 1, index2 = 3. We return [1, 3].

Example 3:

Input: numbers = [_-1_ ,_0_], target = -1 **Output:** [1,2] **Explanation:** The sum of -1 and 0 is -1. Therefore index1 = 1, index2 = 2. We return [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**.

Code Snippets

C++:

```
class Solution {
public:
vector<int> twoSum(vector<int>& numbers, int target) {

}
};
```

Java:

```
class Solution {
public int[] twoSum(int[] numbers, int target) {

}
}
```

Python3:

```
class Solution:
def twoSum(self, numbers: List[int], target: int) -> List[int]:
```