LeetCode 167 (Two Sum II - Input Array is Sorted)

Given 1-indexed array of integers numbers that is already sorted in non-decreasing order, find 2 no. such that they add up to a specific target number. Let these two numbers be numbers[index₁] and number[index₂] where 1<= index₁ < index₂ <= number.length.

Return the indices of the two numbers, index₁ and index₂, added by one as an integer array [index₁, index₂] of length 2.

Constraints

- 2 <= numbers.length <= 3 × 10⁴
- -1000 <= numbers[i] <= 1000
- numbers are sorted in non-decreasing order.
- -1000 <= target <= 1000
- Exactly one solution.

Example

Input ⇒ numbers = [2, 7, 11, 15]
target = 9

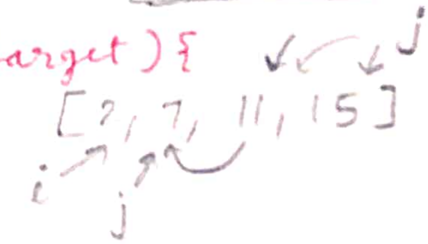
Output ⇒ [1, 2]

Explanation ⇒ The sum of 2 and 7 is 9.

∴ index₁ = 1 } $\xrightarrow{\text{return}}$ [1, 2]
index₂ = 2

Solution :

target = 9



```
class Solution {
  public int[] twoSum(int[] numbers, int target){
    int i = 0; 0
    int j = numbers.length - 1; (4-1)=③
    int[] result = new int[2];
    while (i<= numbers length){
      if (numbers[i] + numbers[j] > target){
         j--;
      }else if (numbers[i] + numbers[j] <target){
         i++;
      }else {
         result [0] = i+1; [0] = 0+1 = 1 } → [1,2]
         result [1] = j+1; [1] = 1+1 = 2
         return result;
      }
    }
    return result; ⟶ [1,2]
  }
}
```