

## 167 Two Sum II - Input array is sorted

2018年4月3日 16:00

### Question:

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

The function twoSum should return indices of the two numbers such that they add up to the target, where index1 must be less than index2. Please note that your returned answers (both index1 and index2) are not zero-based.

You may assume that each input would have *exactly* one solution and you may not use the *same* element twice.

**Input:** numbers={2, 7, 11, 15}, target=9

**Output:** index1=1, index2=2

来自 <<https://leetcode.com/problems/two-sum-ii-input-array-is-sorted/description/>>

给出一个整数数组，返回其中两个数的索引，使得这两个数之和等于指定的目标值。

可以假定每个输入只有一个解，且不能使用一个元素两次。

Example:

Given nums = [2, 7, 11, 15], target = 9,

Because nums[0] + nums[1] = 2 + 7 = 9,

return [0, 1].

来自 <<https://leetcode.com/problems/two-sum/description/>>

### Solution for Python3:

```
1  class Solution(object):
2      def twoSum(self, numbers, target):
3          """
4              :type numbers: List[int]
5              :type target: int
6              :rtype: List[int]
7          """
8          i, j = 0, len(numbers) - 1
9          while i < j:
10             t = numbers[i] + numbers[j]
11             if t < target:
12                 i += 1
13             elif t > target:
14                 j -= 1
15             else:
16                 return [i + 1, j + 1]
17         return None
```

### Solution for C++:

```
1  class Solution {
2  public:
3      vector<int> twoSum(vector<int>& numbers, int target) {
4          int i = 0, j = numbers.size() - 1, sum;
5          while(i < j) {
6              sum = numbers[i] + numbers[j];
7              if( sum == target) {
```

```
8         return vector<int> {i + 1, j + 1};
9     } else if (sum < target) {
10         i++;
11     } else {
12         j--;
13     }
14 }
15 return vector<int> ();
16 }
17 };
```