

# 1. Two Sum

🕒 Created	@July 10, 2020 6:08 AM
▼ Difficulty	Easy
≡ LC Url	<a href="https://leetcode.com/problems/two-sum/">https://leetcode.com/problems/two-sum/</a>
▼ Importance	
≡ Tag	Array&Sorting Hashmap NEET Two pointers
≡ Video	<a href="https://maxming0.github.io/2021/08/20/Two-Sum/">https://maxming0.github.io/2021/08/20/Two-Sum/</a>

Given an array of integers `nums` and an integer `target`, return *indices of the two numbers such that they add up to* `target`.

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

You can return the answer in any order.

## Example 1:

```
Input: nums = [2,7,11,15], target = 9
Output: [0,1]
Explanation: Because nums[0] + nums[1] == 9, we return [0, 1].
```

## Example 2:

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

## Example 3:

```
Input: nums = [3,3], target = 6
Output: [0,1]
```

## Constraints:

- `2 <= nums.length <= 104`
- `-109 <= nums[i] <= 109`
- `-109 <= target <= 109`
- Only one valid answer exists.

## Follow-up:

Can you come up with an algorithm that is less than

$O(n^2)$

time complexity?

## Solution

```
class Solution:
    def twoSum(self, nums: List[int], target: int) -> List[int]:
        prevMap = {} # val -> index

        for i, n in enumerate(nums):
            diff = target - n
            if diff in prevMap:
                return [prevMap[diff], i]
            prevMap[n] = i
```

### 复杂度分析

- 时间复杂度： $O(N)$ ，其中  $N$  是数组中的元素数量。对于每一个元素  $x$ ，我们可以  $O(1)$  地寻找  $target - x$ 。
- 空间复杂度： $O(N)$ ，其中  $N$  是数组中的元素数量。主要为哈希表的开销。

```
class Solution:
    def twoSum(self, nums: List[int], target: int) -> List[int]:
        temp = nums.copy()
        temp.sort()

        start, end = 0, len(nums) - 1

        while start < end:
            if (temp[start] + temp[end]) > target:
                end -= 1
            elif (temp[start] + temp[end]) < target:
                start += 1
            else:
                break

        index1 = nums.index(temp[start])
        # nums.pop(index1)
        index2 = nums.index(temp[end])
        # if index2 >= index1:
        #     index2 += 1
        return [index1, index2]
```

```
class Solution:
    def twoSum(self, nums: List[int], target: int) -> List[int]:
        # 2. 双指针
        # 作者: yun-yu-chen
```

```
# 链接: https://leetcode.cn/problems/two-sum/solution/san-chong-fang-fa-bao-li-shuang-zhi-zhen-ha-xi-san/

temp = nums.copy()
temp.sort()

start, end = 0, len(nums) - 1

while start < end:
    if (temp[start] + temp[end]) > target:
        end -= 1
    elif (temp[start] + temp[end]) < target:
        start += 1
    else:
        break
index1 = nums.index(temp[start])
nums.pop(index1)

index2 = nums.index(temp[end])
if index2 >= index1:
    index2 += 1
return [index1, index2]
```