



Problem List



Dynamic Layout

0



Description

Editorial

Solutions (27.5K)

Submissions

Python3

Auto

Companies

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:**

- $2 \leq \text{nums.length} \leq 10^4$
- $-10^9 \leq \text{nums}[i] \leq 10^9$
- $-10^9 \leq \text{target} \leq 10^9$
- Only one valid answer exists.**

**Follow-up:** Can you come up with an algorithm that is less than  $O(n^2)$  time complexity?

```
1 class Solution:
2     def twoSum(nums, target) -> List[int]:
3         for i in range(len(nums)):
4             for j in range(i+1, len(nums)):
5                 if nums[i] + nums[j] == target:
6                     return [i, j]
7         return []
8
```

Console ^



Run

Submit

Saved to local