Problem 1: Two Sum

Problem Information

Difficulty: Easy
Acceptance Rate: 56.57%
Paid Only: No
Tags: Array, Hash Table

Problem Description

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(n2)` time complexity?

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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