1. Two Sum

Easy 38862 1246 Add to List Share

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 <= 10^4`
- `-10^9 <= nums[i] <= 10^9`
- `-10^9 <= target <= 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?

Java Autocomplete

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

        HashMap<Integer, Integer> numberMap = new HashMap<>();

        for(int i = 0 ; i < nums.length; i++){
            int difference = target - nums[i];

            if(numberMap.containsKey(difference)){
                return new int[]{i, numberMap.get(difference)};
            }

            numberMap.put(nums[i], i);
        }


        return new int[2];
    }
}
```

Testcase Run Code Result Debugger

Accepted Runtime: 0 ms

Your input
```
[2,7,11,15]
9
```

Output `[1,0]` Diff

Expected `[0,1]`