
Problem Statement:

Given an integer array nums and an integer target, find two numbers such that their sum equals the target. Return their indices.

Solution

```python
def two_sum(nums: List[int], target: int) -> List[int]:
    number_indices = {}

    for i in range(len(nums)):
        difference = target - nums[i]
        if (difference in number_indices): return [number_indices[difference], i]
        number_indices[nums[i]] = i

    return []
```

Step-by-Step Breakdown

1. **Input:**
    - nums = [2, 7, 11, 15]
    - target = 9
2. **Hash Map:**
    - Create an empty dictionary *number_indices*.
    - Iterate through each index *i*.
    - Compute *difference = target - nums[i]*.
    - if difference already exists in *number_indices* return *[number_indices[difference], i]*.
    - Otherwise, store nums[i] and its index in number_indices.

    After Processing: number_indices = {2: 0, 7: 1}
3. **Output:** [0, 1]
    - 9 – 2 = 7
    - 7 Does not exist in the dict
        - Add nums[0] and current index to dict
            1. {2: 0, 7: 1}
    - 9 – 2 = 2
    - 2 exists in the dict

- Return [value associated with 2 = 0, current index = 1]

Aspect	Solution One
Efficiency	O(n)
Memory Usage	O(n)
