

덧셈하여 타겟을 만들 수 있는 배열의 두 숫자 인덱스를 리턴하라.

Input: nums = [2,7,11,15], target = 9

Output: [0,1]

### 1. 브루트 포스

```
def twoSum(nums: List[int], target: int) -> List[int]:  
    for i in range(len(nums)):  
        for j in range(i + 1, len(nums)):  
            if nums[i] + nums[j] == target:  
                return [i, j]
```

### 2. 브루트 포스 + in

```
def twoSum(nums: List[int], target: int) -> List[int]:  
    for i, n in enumerate(nums):  
        temp_n = target - n  
        if temp_n in nums[i + 1:]:  
            return [i, nums[i + 1:].index(temp_n) + (i + 1)]
```

### 3. 딕셔너리 활용

```
def twoSum(nums: List[int], target: int) -> List[int]:  
    nums_map = {}  
    for i, num in enumerate(nums):  
        nums_map[num] = i  
    for i, num in enumerate(nums):  
        if target - num in nums_map and i != nums_map[target - num]:  
            return [nums.index(num), nums_map[target - num]]
```

```
def twoSum(nums: List[int], target: int) -> List[int]:  
    nums_map = {}  
    for i, num in enumerate(nums):  
        if target - num in nums_map:  
            return [nums_map[target - num], i]  
        nums_map[num] = i
```