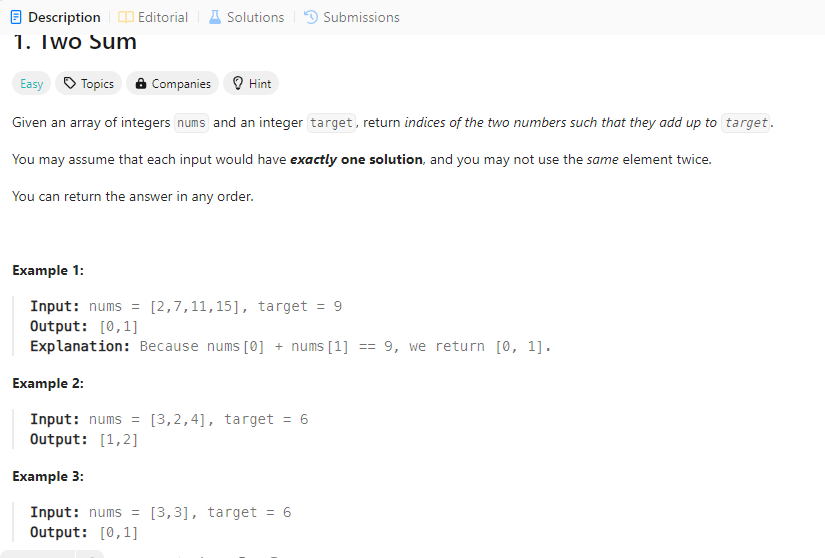
Problem Statement: <https://leetcode.com/problems/two-sum/description/>



## Sol1:

Bruteforce Method:

class Solution:

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

for i in range(len(nums)):

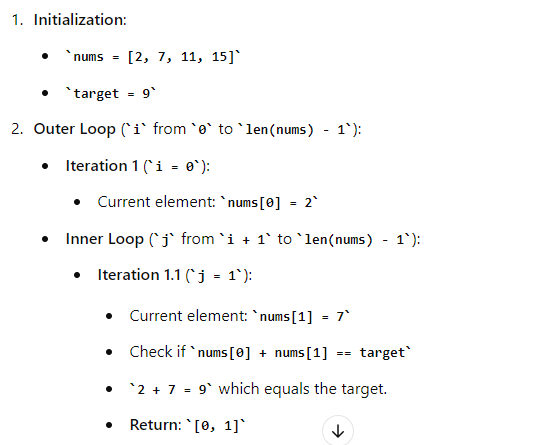
for j in range(i + 1, len(nums)):

if nums[i] + nums[j] == target:

return [i, j]

Time complexity: O(n^2)

## Dry Run:



### Single Pass Method:

class Solution:

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

hashmap = {}

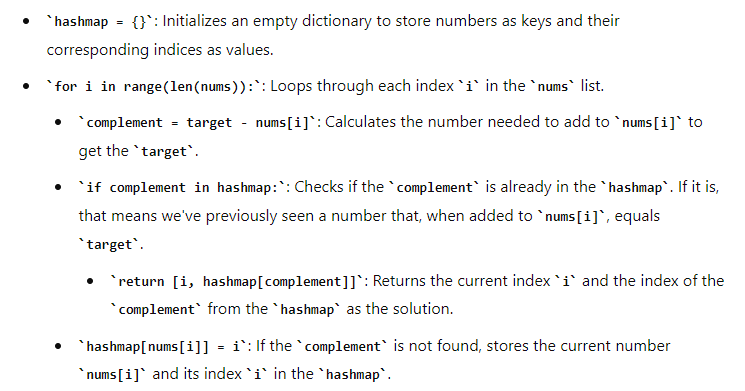
for i in range(len(nums)):

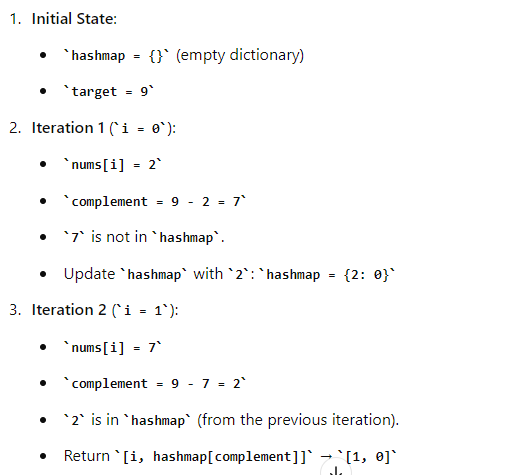
complement = target - nums[i]

if complement in hashmap:

return [i, hashmap[complement]]

hashmap[nums[i]] = i





Complexity : O(n)

### Two Pass method:

class Solution:

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

hashmap = {}

# First pass: build the hash table

for i in range(len(nums)):

hashmap[nums[i]] = i

# Second pass: check for complement

for i in range(len(nums)):

complement = target - nums[i]

if complement in hashmap and hashmap[complement] != i:

return [i, hashmap[complement]]