1. TWO SUM

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

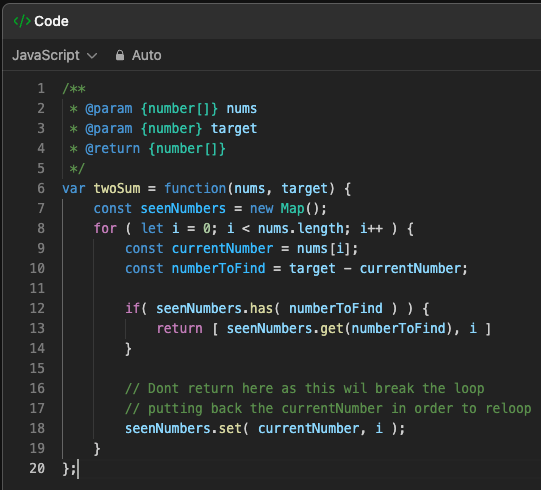
-109 <= nums[i] <= 109

-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?

SOLUTION:



1. About the doc of the problem

* Nums is the array of numbers like [1,2,3,4]
* Target is the single number
* Function returns an array of two numbers i.e. indices of the two numbers in nums that add up to the target

1. Created a map to store the seen numbers



* This will store the key value pairs as the numbers and the index respectively.

1. Now the entire logic from here is straight forward.

* We loop through all the numbers in the array
* We check if the pair exists
* Otherwise store the current number for the future and all set.