

## \* Two Sum :-

- Each input will have exactly 1 solution.
- Same element cannot be used twice.

## \* Brute force :-

- Use nested loops to iterate over the array twice.
- Check if  $\text{nums}[i] + \text{nums}[j] == \text{target}$ .
- If yes, then return  $[i, j]$

Time complexity :-  $O(n^2)$

Space complexity :-  $O(1)$

## \* Using Hash tables :-

$\text{nums} = [3, 2, 4]$        $\text{target} = 6$

- Create an empty map that will map  $\text{nums}[i]$  with  $i$ .
- Iterate over the array & calculate the complement =  $\text{target} - \text{nums}[i]$ .
- Check if this complement exists in the map.
  - ↳ Yes → return  $[i, \text{index of complement}]$
  - ↳ No → Add (current Element,  $i$ ) to map

$TC = SC = O(N)$

**Code** - [https://github.com/TheParthMaru/DSA/tree/main/leetcode/0001\\_Two\\_Sum](https://github.com/TheParthMaru/DSA/tree/main/leetcode/0001_Two_Sum)

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