724. Find Pivot Index

Given an array of integers `nums`, calculate the **pivot index** of this array.

The **pivot index** is the index where the sum of all the numbers **strictly** to the left of the index is equal to the sum of all the numbers **strictly** to the index's right.

If the index is on the left edge of the array, then the left sum is `0` because there are no elements to the left. This also applies to the right edge of the array.

Return *the **leftmost pivot index***. If no such index exists, return -1.

Example 1:

```
Input: nums = [1,7,3,6,5,6]
Output: 3
Explanation:
The pivot index is 3.
Left sum = nums[0] + nums[1] + nums[2] = 1 + 7 + 3 = 11
Right sum = nums[4] + nums[5] = 5 + 6 = 11
```

Example 2:

```
Input: nums = [1,2,3]
Output: -1
Explanation:
There is no index that satisfies the conditions in the problem statement.
```

Example 3:

```
Input: nums = [2,1,-1]
Output: 0
Explanation:
The pivot index is 0.
Left sum = 0 (no elements to the left of index 0)
Right sum = nums[1] + nums[2] = 1 + -1 = 0
```

Constraints:

- `1 <= nums.length <= 10<sup>4</sup>`
- `-1000 <= nums[i] <= 1000`

-
  ```
  class Solution:
      def pivotIndex(self, nums: List[int]) -> int:
          prefix = 0
          forward  = [0]*len(nums)
          backward = [0]*len(nums)

          for i in range(len(nums)):
              prefix+=nums[i]
              forward[i] = prefix
          prefix = 0
          for i in range(len(nums)-1,-1,-1):
              prefix+=nums[i]
              backward[i] = prefix

          for i in range(len(nums)):
              if i==0:
                  if backward[i+1]==0:
                      return 0
              elif i==len(nums)-1:
                  if forward[i-1]==0:
                      return len(nums)-1
              else:
                  if forward[i-1]==backward[i+1]:
                      return i
          return -1
  ```