

Find Pivot Index

Given an array of integers `nums`, write a method that returns the "pivot" index of this array.

We define the pivot index as the index where the sum of the numbers to the left of the index is equal to the sum of the numbers to the right of the index.

If no such index exists, we should return -1. If there are multiple pivot indexes, you should return the left-most pivot index.

Example 1:

Input:

`nums = [1, 7, 3, 6, 5, 6]`

Output: 3

Explanation:

The sum of the numbers to the left of index 3 (`nums[3] = 6`) is equal to the sum of numbers to the right of index 3.

Also, 3 is the first index where this occurs.

Example 2:

Input:

`nums = [1, 2, 3]`

Output: -1

Explanation:

There is no index that satisfies the conditions in the problem statement.

Note:

- The length of `nums` will be in the range `[0, 10000]`.
- Each element `nums[i]` will be an integer in the range `[-1000, 1000]`.

Solution 1

```
class Solution {  
    public int pivotIndex(int[] nums) {  
        int sum = 0, left = 0;  
        for (int i = 0; i < nums.length; i++) sum += nums[i];  
  
        for (int i = 0; i < nums.length; i++) {  
            if (i != 0) left += nums[i - 1];  
            if (sum - left - nums[i] == left) return i;  
        }  
  
        return -1;  
    }  
}
```

written by [shawngao](#) original link [here](#)

Solution 2

As we iterate through the array of numbers, we need to keep track of the sum of the values on the current number's left and its right. The following debugger trace demonstrates the values of the variables in each loop before the `left == right` line

Input: [1, 7, 3, 6, 5, 6]

1. index: 0, num: 1, left: 0, right: 27
2. index: 1, num: 7, left: 1, right: 20
3. index: 2, num: 3, left: 8, right: 17
4. index: 3, num: 6, left: 11, right: 11 <-- Found!!!

- Yangshun

```
class Solution(object):
    def pivotIndex(self, nums):
        # Time: O(n)
        # Space: O(1)
        left, right = 0, sum(nums)
        for index, num in enumerate(nums):
            right -= num
            if left == right:
                return index
            left += num
        return -1
```

written by [yangshun](#) original link [here](#)

Solution 3

Java

```
class Solution {  
    public int pivotIndex(int[] nums) {  
        int total = 0, sum = 0  
        for (int num : nums) total += num;  
        for (int i = 0; i < nums.length; sum += nums[i++])  
            if (sum * 2 == total - nums[i]) return i;  
        return -1;  
    }  
}
```

C++

```
class Solution {  
public:  
    int pivotIndex(vector<int>& nums) {  
        int total = 0;  
        for (int num : nums) total += num;  
        int sum = 0;  
        for (int i = 0; i < nums.size(); sum += nums[i++])  
            if (sum * 2 == total - nums[i])  
                return i;  
  
        return -1;  
    }  
};
```

written by [alexander](#) original link [here](#)

From [Leetcode](#).