# Equilibrium Index

Find an index in an array such that the sum of elements to the left is equal to the sum of elements to the right.

### Equilibrium Index in an Array

An **Equilibrium Index** is an index in an array where:

**Sum of elements on the left = Sum of elements on the right**

👉 The element at that index is **NOT included** in either sum.

---

### 📌 Definition

For an array `arr[]`, an index `i` is an equilibrium index if:

```
arr[0] + arr[1] + ... + arr[i-1]
=
arr[i+1] + arr[i+2] + ... + arr[n-1]
```

### Efficient Approach (Using Prefix Logic)

Instead of calculating left and right sums repeatedly:

## Steps:

1. Find total sum of array.
2. Traverse the array.
3. Keep updating left sum.
4. Right sum = total sum − left sum − current element.
5. Compare both.

## Java Code Implementation

```
class EquilibriumIndex {
    public static void main(String[] args) {
        int[] arr = {1, 3, 5, 2, 2};
        int n = arr.length;

        int totalSum = 0;

        // Step 1: Calculate total sum
        for (int num : arr) {
            totalSum += num;
        }

        int leftSum = 0;

        // Step 2: Find equilibrium index
        for (int i = 0; i < n; i++) {

            totalSum -= arr[i]; // now totalSum becomes right sum

            if (leftSum == totalSum) {
                System.out.println("Equilibrium Index = " + i);
                return;
            }

            leftSum += arr[i];
        }

        System.out.println("No Equilibrium Index found");
    }
}
```

---

## ⏱️ Time Complexity

| Method | Time |
|---|---|
| Brute Force | $O(n^2)$ |
| Efficient Method | **O(n)** |

## 🎯 Key Points to Remember

• There can be **more than one** equilibrium index
• First and last index can also be equilibrium (if one side sum = 0)
• Works with negative numbers too