Linear Search

Linear Search is a simple search algorithm used to find a specific element in an array. It checks each element of the array one by one until the target value is found or the end of the array is reached.

Examples

Example 1:

```
Input: arr = [2, 4, 7, 10], target = 10
   Output: 3
   Explanation: 10 is found at index 3
```

Example 2:

```
Input: arr = [6, 8, 0, 3], target = 5
   Output: -1
   Explanation: 5 is not present in the array
```

Approach:

Start from the first element of the array.
Compare the current element with the target value.
If a match is found, return the index.
If the loop ends without finding the target, return `-1`.

Dry Run:

Input:
Array: [4, 5, 1, 3, 9]
Target: 5
i = 0: arr[0] = 4 → Not equal to 5 → Continue
i = 1: arr[1] = 5 → Equal to 5 → **Return 1**

Output: Element found at index `1`

Time Complexity (TC):

– In the worst case, the algorithm traverses the entire array.
– Each element is checked exactly once.
TC = O(n), where `n` is the size of the array.

Space Complexity (SC):

– The algorithm does not use any extra space.
SC = O(1) (constant space)

JavaScript | C++ | C | Java | Python

```javascript
let arr = [4, 5, 1, 3, 9];

function linearSearch(arr, target) {
  for (let i = 0; i < arr.length; i++) {
    if (arr[i] == target) {
      return i;
    }
  }
  return -1;
}

let result = linearSearch(arr, 5);
console.log("Element found at index", result);
```