



# Linear Search

## 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.

### 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.

### Time & Space Complexity:

**Time Complexity:**  $O(n)$  where  $n$  is the size of the array.

In the worst case, the algorithm traverses the entire array.

Each element is checked exactly once.

**Space Complexity:  $O(1)$**  Constant Space

## Dry Run

**Input:** arr = [4, 5, 1, 3, 9], target = 5

i = 0, arr[i] = 4

→ 4 == 5 ? No

i = 1, arr[i] = 5

→ 5 == 5 ? Yes → return 1

Loop breaks here since target found.

**Output:** Element found at index 1

JavaScript

Python

Java

C++

C

C#

```
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);
```

**Video**

**Course**

**Discuss doubts**

**Contribute**

**Certificate**



## Linear Search - DSA Notes

Linear Search - DSA Notes