Linear Search

Linear Search (also called Sequential Search) is the most basic searching algorithm.It goes element by element through a list until it finds the target or reaches the end.

Think of it like this

> Flipping through a notebook page by page to find a specific note, no skipping ahead, just checking one at a time.

🔹 Algorithmic Steps (in pseudocode / algorithm format)

```
Algorithm: LinearSearch(A, n, key)

Input: Array A of n elements, value 'key' to be searched

Output: Index of key if found, else -1


1. for i ← 0 to n-1 do

2.     if A[i] == key then

3.         return i       // key found at position i

4. end for

5. return -1              // key not found
```

That's it; dead simple. No sorting, no fancy math, just brute-force comparison.

Implementation in **C++**

```
#include <iostream>

using namespace std;


int linearSearch(int arr[], int n, int key) {

    for (int i = 0; i < n; i++) {

        if (arr[i] == key) {

            return i;  // return index where key is found

        }

    }

    return -1;  // not found

}
```

Time Complexity

Case	Comparisons	Time
Best (key at first position)	1	O(1)
Average	n/2	O(n)
Worst (key absent or last element)	n	O(n)

Space complexity: O(1), no extra memory used.
Sorted array needed? ❌ No. Works on **unsorted data.**
Stable? ✅ Not really applicable here, since it's not a sorting algorithm.