

Linear Search Algorithm

Linear search is a very basic and simple search algorithm. In Linear search, we search an element or value in a given array by traversing the array from the starting, till the desired element or value is found.

As we learned in the [previous tutorial](#) that the time complexity of Linear search algorithm is **O(n)**, we will analyse the same and see why it is **O(n)** after implementing it.

Implementing Linear Search

Following are the steps of implementation that we will be following:

1. Traverse the array using a **for** loop.
2. In every iteration, compare the **target** value with the current value of the array.
 - If the values match, return the current index of the array.
 - If the values do not match, move on to the next array element.
3. If no match is found, return **-1**.

To search the number **5** in the array given below, linear search will go step by step in a sequential order starting from the first element in the given array.

8	2	6	3	5
---	---	---	---	---

```
/*  
  
    below we have implemented a simple function  
    for linear search in C  
  
    - values[] => array with all the values  
    - target => value to be found  
    - n => total number of elements in the array  
*/  
  
int linearSearch(int values[], int target, int n)  
{  
  
    for(int i = 0; i < n; i++)  
  
        {
```

```
        if (values[i] == target)

            {

                return i;

            }

    }

    return -1;

}
```

Copy

Some Examples with Inputs

Input: values[] = {5, 34, 65, 12, 77, 35}

target = 77

Output: 4

Input: values[] = {101, 392, 1, 54, 32, 22, 90, 93}

target = 200

Output: -1 (not found)

Final Thoughts

We know you like Linear search because it is so damn simple to implement, but it is not used practically because binary search is a lot faster than linear search. So let's head to the next tutorial where we will learn more about binary search.