25 Oct 2025 Date

Linear Search

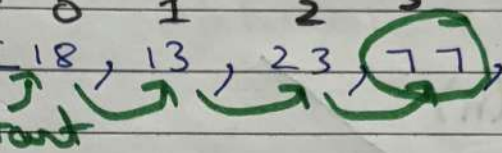
Searching : It is a process of finding a given position in a list of values.

Linear / Sequential Search :

- It is basic of simple search algorithm.
- In sequential search, we compare the target value with all the other elements given in the list.

e.g:- arr = [18, 13, 23, 77, 8, 0] (Unsorted Array)

target = 77 start



In above example, the target value is compared with all the elements in array in sequential/linear array.

Time Complexity →

→ **Best Case :** O(1) → Constant

⇒ How many checks will the loop make in best case i.e the element will be found at 0th index. i.e only one comparison will be made for best case.

It does not matter weather the array size is of 100 elements or 10^5 element, we will only make one comparison as our target is on 0th index. So it is having constant time complexity.

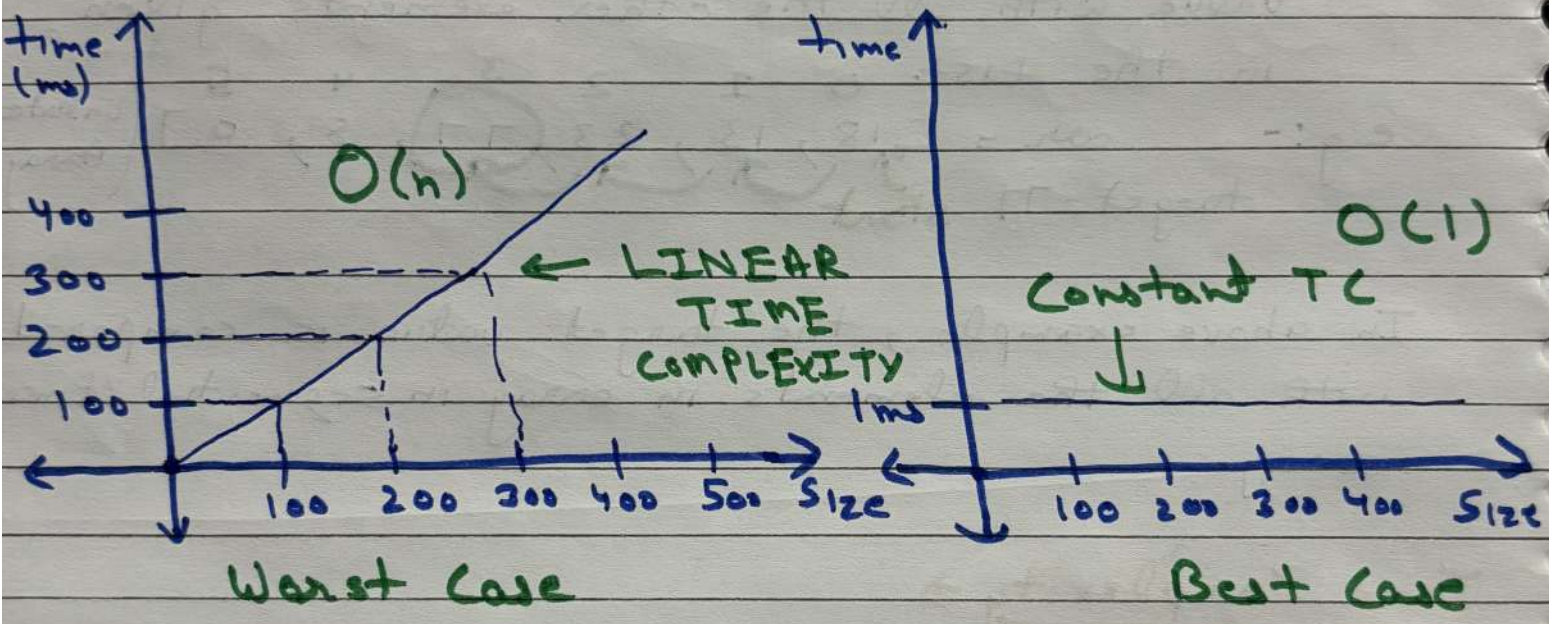
Date

→ Worst Case : O(n)

⇒ Worst case, here it will go through every element and then it says element not found.

⇒

Size of Array	No of comparisons	Time (ms)
100	100	100 ms
200	200	200 ms
n	n	n ms



Space Complexity → It is extra space taken to solve the problem.

In case of linear search, we only took one variable to run the loop.

Size of array	Space Taken
100	1
200	1
n	1

Best Case → O(1) Worst Case → O(1)

Date

Search in a string →

'Shubham'
Target = b

⇒ Internally, strings are using arrays only. So we can get the character using index and also we can loop over string, same as arrays.

```
          0 1 2 3 4 5 6
str = 'Shubham'
       ↑
      start
```

How to access character of string → str[idx]
str.charAt(idx)

TC → O(N)
SC → O(1)

Search in a range → Given start and end

Normally when we search in the given array, we starts from 0 and check every element till the end. So in this case, start = 0 & end = arr.length - 1

But here, we are taking start & end as input from user so we can check in subarray.
In worst case, user can provide start = 0 & end = n-1

```
arr = [ 5, 2, 14, 10, 12 ]
        ↑          ↑
      start      end
```

Target = 15 , Result = -1 (Target Element not found)

Date

Search in 2d array → Matrix given (m x n)

m → Row
n → Col

Row \ Col	0	1	2	3
0	[10,	12,	8]	
1	[5,	15,	10,	8]
2	[42,	31,	18,	16]

2d array is an array inside array.

Outside array length is number of rows. Each row can have different number of elements which will be the number of cols for each row.

Traverse 2d Array

Row \ Col	0	1	2	3
0	[10,	12,	8]	
1	[5,	15,	10,	8]
2	[42,	31,	18,	16]

Row

Loop → 0 [0,0] [0,1] [0,2]
1 [1,0] [1,1] [1,2] [1,3]
2 [2,0] [2,1] [2,2] [2,3]

Date

Outer loop will run m times (Rows)
Inner loop will run n times (cols)

$$TC \rightarrow O(m \times n)$$
$$SC \rightarrow O(1)$$

Worst Case → Element is not found in matrix even after traversing the complete 2d array.

If m is 3 & n is 4 so we do the comparison of 3 x 4 times = 12 times.
That's why TC → O(mxn)

If no element found return (-1, -1) as -1 can never be the index.