Linear Search

arr = [18, 12, 9, 14, 77, 50]
Find whether 14 exists in array or not?

→ By for each loop

0i 1i 2i (3)i 4 5
arr = [18, 12, 9, 14, 77, 50]

Let's say if we have to search for 88 in the same array then again it will iterate over each and every element. The Element 88 will not be found and you will reach at the last index. In this case, if value not found, return -1.

Time Complexity →

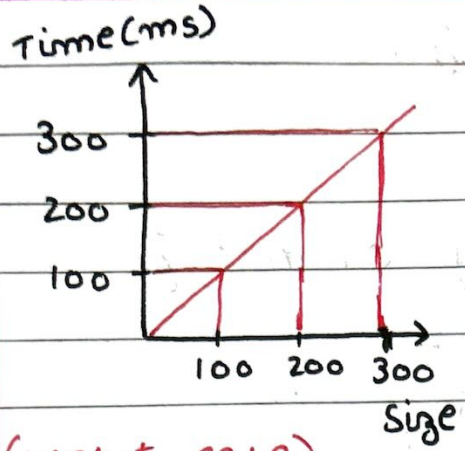
Best → O(1) // Constant	Worst Case → O(N) N → Size of Array

How many checks will the loop make in best case i.e element found at 0th index?

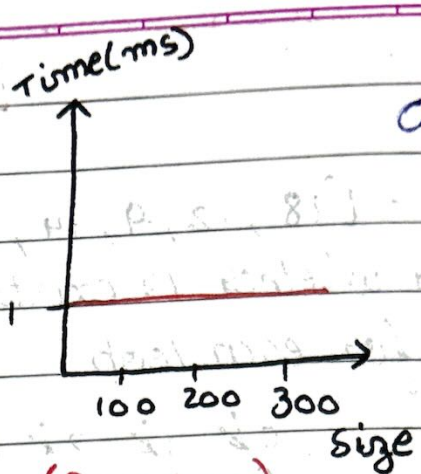
(0)i 1 2 3
arr = [8, 9, 12, 18 --- 200 Elements]
target = 8 1 Comparison in the best case

Worst case → You don't find the target item. iterate / go through every item and at the end it says I did not find it.

Size of Array = 100 → 100 Comparisons = 100 ms
200 → 200 Comparison = 200 ms
1 Lakh → 1 Lakh Comparison = 1 Lakh ms



(worst case)
Linear Time
Complexity



(Best case)
Constant Time
Complexity

Search in 2-D Arrays

```
[[1, 2, 3]
 [9, 8, 5]
 [6, 19, 4]]
```

```
for (row=0; row < len(arr);
     row++)
{
  for(c=0; c < len(row); c++)
  {
    if(arr[r][c] ==
         target)
    { // Element found
    }
  }
}
```