

Complexity Analysis of Linear Search:

Time Complexity:

- **Best Case:** In the best case, the key might be present at the first index. So the best case complexity is $O(1)$
- **Worst Case:** In the worst case, the key might be present at the last index i.e., opposite to the end from which the search has started in the list. So the worst-case complexity is $O(N)$ where N is the size of the list.
- **Average Case:** $O(N)$

Auxiliary Space: $O(1)$ as except for the variable to iterate through the list, no other variable is used.

Advantages of Linear Search:

- Linear search can be used irrespective of whether the array is sorted or not. It can be used on arrays of any data type.
- Does not require any additional memory.
- It is a well-suited algorithm for small datasets.

Drawbacks of Linear Search:

- Linear search has a time complexity of $O(N)$, which in turn makes it slow for large datasets.
- Not suitable for large arrays.

When to use Linear Search?

- When we are dealing with a small dataset.
- When you are searching for a dataset stored in contiguous memory.