Linear Search

Linear search is the simplest search algorithm. It checks each element one by one. This is very simple and easy to understand and implement.

To implement the Linear Search algorithm we need:

1. An array with values to search through.
2. A target value to search for.
3. A loop that goes through the array from start to end.
4. An if-statement that compares the current value with the target value and returns the current index if the target value if found.
5. After the loop, return -1, because at this point we know the target value has not been found.

It works:

1. Go through the array value by value from the start.
2. Compare each value to check if it is equal to the value we are looking for.
3. If the value is found, return the index of that value.
4. If the end of the array is reached and the value is not found, return -1 to indicate that the value was not found.

Linear Search Time Complexity

Best Case:

- Target is found in the **first element**
- Only **1 comparison**
- **Time complexity: O(1)**

Average Case:

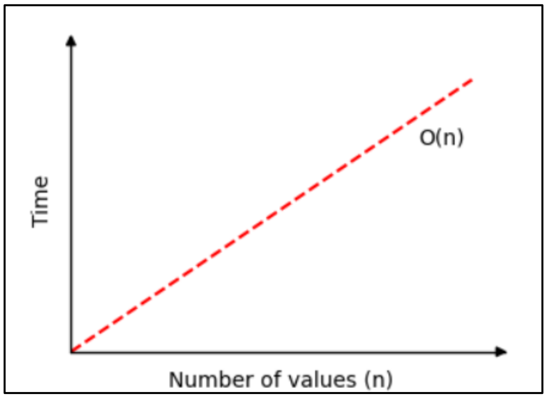
- Assume target may appear anywhere randomly
- On average, Linear Search checks **n/2 elements**
- Constants are ignored in Big-O, so
  Average case = O(n)

Since Big-O focuses on **worst-case performance**, Linear Search behaves as:

Time Complexity = O(n)

Worst Case:

- Target is **not** found, or found at the **last** index
- Needs to check **all n elements**
- **n comparisons**
- **Time complexity: O(n)**


Time
O(n)
Number of values (n)