

## Compare the time complexity of linear and binary search algorithms:

### Linear Search:

- **Best Case ( $O(1)$ ):** This happens when the element we're looking for is right at the beginning of the list.
- **Average Case ( $O(n)$ ):** Usually, we might have to check about half the items before finding it.
- **Worst Case ( $O(n)$ ):** If the element is at the very end or not in the list at all, we have to go through the entire list.

### Binary Search:

- **Best Case ( $O(1)$ ):** This happens when the element is exactly in the middle of the sorted list.
- **Average Case ( $O(\log n)$ ):** Every time we search, the list is split in half, which makes it much faster.
- **Worst Case ( $O(\log n)$ ):** Even in the worst case, the number of steps is far less than linear search.

---

## Discussing which algorithm is more suitable for E-Commerce platform

For an **e-commerce platform**, where there are huge numbers of products, binary search is a good choice as long as the product list is sorted.

- It's faster than linear search, specially when you're dealing with thousands of items.
- Binary search uses the **divide and conquer** technique to reduce search range by half every time, which is an efficient way.
- Its time complexity of  $O(\log n)$  makes it suitable for real-world platforms where speed matters.