**Linear Search**

**Description**: Linear Search is a simple search algorithm that sequentially checks each element in a list until the desired element is found or the end of the list is reached.

**Algorithm**:

1. Start at the beginning of the list.
2. Compare the target value with the current element.
3. If they match, return the index of the current element.
4. If they do not match, move to the next element.
5. Repeat until the target is found or the end of the list is reached.
6. If the target is not found, return an indicator that the element is not present (e.g., -1).

**Binary Search**

**Description**: Binary Search is an efficient search algorithm that works on sorted lists by repeatedly dividing the search interval in half. It compares the target value to the middle element and narrows down the search to the appropriate half.

**Algorithm**:

1. Start with two pointers: low (beginning of the list) and high (end of the list).
2. Find the middle element of the current range.
3. Compare the middle element with the target value.
4. If they match, return the index of the middle element.
5. If the target value is less than the middle element, narrow the search to the left half by updating high.
6. If the target value is greater than the middle element, narrow the search to the right half by updating low.
7. Repeat steps 2-6 until the target is found or low exceeds high.
8. If the target is not found, return an indicator that the element is not present (e.g., -1).

**Time Complexity Comparison**

**Linear Search:**

- Best Case: O(1) (target is the first element)

- Average Case: O(n) (target is somewhere in the middle)

- Worst Case: O(n) (target is at the end or not present)

**Binary Search:**

- Best Case: O(1) (target is the middle element)

- Average Case: O(log n) (list is repeatedly halved)

- Worst Case: O(log n) (target is not present but requires checking all possible divisions)

**When to Use Each Algorithm**

**Linear Search:**

- Use When:

- The list is unsorted or unordered.

- The dataset is small or infrequently searched.

- Simplicity and ease of implementation are priorities.

**Binary Search:**

- Use When:

- The list is sorted or can be sorted.

- The dataset is large and efficient searching is crucial.

- You need fast search times and the overhead of sorting is acceptable.

**Summary:**

- Linear Search is suitable for small or unsorted datasets due to its simplicity.

- Binary Search is preferable for large, sorted datasets due to its efficiency.