**Linear Search**:

* Linear search checks each element in the list sequentially until it finds the target element or reaches the end of the list.
* Time Complexity: O(n), where n is the number of elements in the list.
* It does not require the list to be sorted.

**Binary Search**:

* Binary search works on sorted lists. It repeatedly divides the list in half, comparing the target element with the middle element of the list.
* If the target is equal to the middle element, the search is complete.
* If the target is less than the middle element, it searches the left half of the list.
* If the target is greater than the middle element, it searches the right half of the list.
* Time Complexity: O(log n), where n is the number of elements in the list.
* The list must be sorted for binary search to work.

**When to Use Each Algorithm**:

* **Linear Search**:
  + Useful when the list is small or unsorted.
  + Suitable for lists where the overhead of sorting is not justified by the search speedup.
* **Binary Search**:
  + Ideal for large, sorted lists.
  + Requires the list to be sorted, so there is an initial overhead for sorting