**Time Complexity Comparison:**

* **Linear Search**
  + **Best Case: O(1) (first element match)**
  + **Average/Worst Case: O(n)**
  + **Each element is checked one by one until the target is found or end of array is reached.**
* **Binary Search**
  + **Best Case: O(1) (middle element match)**
  + **Average/Worst Case: O(log n)**
  + **The array must be sorted before performing binary search.**
  + **Each step eliminates half of the remaining elements.**

**When to Use Each Search Algorithm:**

* **Linear Search:**
  + **Works on both sorted and unsorted data.**
  + **Useful for small data sets where sorting overhead isn't justified.**
  + **Simple to implement and does not require additional setup.**
* **Binary Search:**
  + **Requires the data to be sorted in advance.**
  + **Preferred for large datasets due to its logarithmic time performance.**
  + **Very efficient when repeated searches are needed on a sorted list.**