Answers

1. **Linear Search**: Linear search is a simple search algorithm that checks each element in a list sequentially until the desired element is found or the end of the list is reached. It does not require the list to be sorted. The time complexity of linear search is O(n), where n is the number of elements in the list.

**Binary Search**: Binary search is a more efficient algorithm that works on sorted lists. It repeatedly divides the list in half, comparing the target value to the middle element. If the target is equal to the middle element, the search is complete. If the target is less than the middle element, the search continues in the left half; otherwise, it continues in the right half. The time complexity of binary search is O(log n), making it much faster than linear search for large datasets.

4. **Analysis**

**Time Complexity Comparison**

* **Linear Search**: O(n)
* **Binary Search**: O(log n)

**Discussion on When to Use Each Algorithm**

* **Linear Search**: Suitable for small datasets or unsorted lists where the overhead of sorting is not justified. It's straightforward and easy to implement.
* **Binary Search**: Best for large datasets that are already sorted. The logarithmic time complexity allows for significantly faster search times compared to linear search, especially as the dataset grows.