Aswers

1.**Big O Notation**

Big O notation is a mathematical representation used to describe the performance characteristics of an algorithm, specifically its time complexity. It provides an upper bound on the running time of an algorithm in terms of the size of the input data. This helps in understanding how the algorithm will scale as the size of the input increases.

Common complexities in Big O notation include:

* **O(1)**: Constant time
* **O(log n)**: Logarithmic time
* **O(n)**: Linear time
* **O(n log n)**: Linearithmic time
* **O(n²)**: Quadratic time

**Search Operations Scenarios**

1. **Best Case**: The minimum number of comparisons needed to find the desired item. For example, in a linear search, if the desired item is the first element in the array, the search completes in O(1) time.
2. **Average Case**: The expected number of comparisons needed to find the item if the search is performed multiple times under random conditions. For linear search, this is O(n/2), which simplifies to O(n).
3. **Worst Case**: The maximum number of comparisons needed, often when the desired item is the last one or not present at all. For both linear and binary search, the worst case is O(n) for linear search and O(log n) for binary search.

4. **Analysis**

**Time Complexity Comparison**

* **Linear Search**:
  + Best Case: O(1)
  + Average Case: O(n)
  + Worst Case: O(n)
* **Binary Search**:
  + Best Case: O(1)
  + Average Case: O(log n)
  + Worst Case: O(log n)

**Which Algorithm is More Suitable?**

Binary search is generally more suitable for an e-commerce platform due to its logarithmic time complexity, especially when dealing with large datasets.