**E-commerce Platform Search Function**

1. Explain Big O notation and how it helps in analyzing algorithms

* Big O notation describes the upper bound of an algorithm's time complexity, indicating how the runtime or space requirements grow with input size.
* Linear Search:

1. Best Case: O(1) – found at the first position.
2. Average Case: O(n) – found after scanning half of the list.
3. Worst Case: O(n) – found after scanning the entire list or not found.

* Binary Search:

1. Best Case: O(1) – found at the midpoint.
2. Average Case: O(log n) – divided search interval.
3. Worst Case: O(log n) – not found after log n iterations.
4. Compare the time complexity of linear and binary search algorithms

* Linear Search:

1. Time Complexity: O(n) for all cases.
2. Advantages: Simple to implement, works on unsorted data.
3. Disadvantages: Inefficient for large datasets due to linear growth with input size.

* Binary Search:

1. Time Complexity: O(log n) for best, average, and worst cases, provided the data is sorted.
2. Advantages: More efficient than linear search for large datasets, but requires the data to be sorted.
3. Disadvantages: Requires sorting the data, which can be an additional overhead if the data changes frequently.
4. Which Algorithm Is More Suitable

* Binary Search is more suitable for large datasets where the data is sorted or can be sorted. It provides significantly faster search times compared to linear search.
* Linear Search is more appropriate for smaller datasets or unsorted data where sorting is not feasible.