E-commerce Platform Search Function

Understand Asymptotic Notation::

Q1: Explain Big O notation and how it helps in analyzing algorithms.

Ans:

Big O notation describes the worst-case time or space complexity of algorithms, helping to compare their efficiency and scalability by indicating how performance changes with input size.

Q2: Describe the best, average, and worst-case scenarios for search operations.

Ans:

Best-case: The desired element is found immediately, resulting in constant time complexity, O(1).

Average-case: The element is found after searching a typical portion of the dataset, often resulting in O(n) for linear search and O(log n) for binary search.

Worst-case: The element is not present or is found after examining all possible elements, resulting in O(n) for linear search and O(log n) for binary search.

Analysis:

Q1: Compare the time complexity of linear and binary search algorithms.

Ans:

Linear Search:

- Best-case: O(1) (found at the first position)
- Average-case: O(n) (element found after checking half the elements on average)
- Worst-case: O(n) (element not present or found at the end)

Binary Search:

- Best-case: O(1) (found at the middle position)
- Average-case: O(log n) (element found after repeatedly halving the search space)
- Worst-case: O(log n) (element not present, but still requires full log(n) depth search)

Binary search is more efficient than linear search, but requires the dataset to be sorted.

Q2: Discuss which algorithm is more suitable for your platform and why.

Ans:

For a platform with large and frequently queried datasets, binary search is more suitable due to its O(log n) time complexity, offering faster searches compared to linear search's O(n). However, binary search requires data to be sorted.