**Big O Notation:** It is a mathematical notation used to describe the upper bound of an algorithm's running time or space requirements in terms of the size of the input data. Big O notation helps in comparing the efficiency of different algorithms, regardless of hardware or other implementation details. It focuses on the most significant factors affecting performance, ignoring lower-order terms and constant factors.

**Search Operations Scenarios:**

* **Linear Search:**
  + **Best-Case: O(1) (element found at the beginning)**
  + **Average-Case: O(n) (element found halfway through)**
  + **Worst-Case: O(n) (element found at the end or not found)**
* **Binary Search:**
  + **Best-Case: O(1) (element found at the first middle check)**
  + **Average-Case: O(logn) (due to halving the search space)**
  + **Worst-Case: O(logn) (element not present, requiring maximum comparisons)**

**Time Complexity Comparison:**

* **Linear Search:** O(n) for worst and average cases.
* **Binary Search:** O(logn) for worst and average cases

.

**Suitable Algorith for Platform:** Binary search is more suitable for e-commerce search functionality because it's faster O(log n) and works well with sorted data, which is common in e-commerce platforms. It works better for larger datasets.

Linear search is slower O(n) and better for unsorted data, but this is less typical in e-commerce scenarios. It can work good for small data set.