1. **Understand Asymptotic Notation:**

* Explain Big O notation and how it helps in analyzing algorithms.
* Describe the best, average, and worst-case scenarios for search operations.

Answer:-

Big O notation is used to describe the **efficiency** of an algorithm in terms of **time** or **space complexity** as the input size increases. It helps analyze:

* **Best case**: Minimum time taken
* **Average case**: Expected time for random inputs
* **Worst case**: Maximum time taken

**Best, Average, and Worst Case in Searching**

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm** | **Best Case** | **Average Case** | **Worst Case** |
| **Linear Search** | O(1) (first item) | O(n/2) ≈ O(n) | O(n) (last item) |
| **Binary Search** | O(1) (middle item) | O(log n) | O(log n) |

* **Best Case**: Target is found in the first step.
* **Average Case**: Expected number of steps for a random target.
* **Worst Case**: Target is not present, or found in last comparisons.

**4. Analysis:**

* Compare the time complexity of linear and binary search algorithms.
* Discuss which algorithm is more suitable for your platform and why.

Answer:-

|  |  |  |
| --- | --- | --- |
| **Feature** | **Linear Search** | **Binary Search** |
| **Algorithm Type** | Sequential Search | Divide and Conquer |
| **Data Requirement** | Works on **unsorted** or sorted data | Requires **sorted** data |
| **Performance on Large Data** | Slower | Much faster |
| **Flexibility** | Can be used in dynamic and unsorted datasets | Only effective with static and sorted datasets |
| **Search Pattern** | Checks each item one-by-one | Repeatedly divides the array in half |

For an optimized and scalable e-commerce platform, **Binary Search** is more suitable **if the product list is sorted**. It significantly reduces search time for large datasets.