**E-commerce Platform Search Function**

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

* Big O helps in understanding how efficient an algorithm is as the input size increases.
* It ignores small differences and focuses on the general trend as data grows.
* For example:
  + O(1): Constant time (fastest)
  + O(n): Linear time (grows with input)
  + O(log n): Logarithmic (very efficient)

1. **Describe the best, average, and worst-case scenarios for search operations.**

* **Best Case**: Minimum steps needed (e.g., first match).
* **Average Case**: Typical performance over many inputs.
* **Worst Case**: Maximum steps (e.g., not found at all).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Search Type** | **Best Case** | **Average Case** | **Worst Case** | **When to Use** |
| **Linear Search** | |  | | --- | |  |   O(1) | O(n) | O(n) | Small/unsorted list |
| **Binary Search** | O(1) | O(log n) | O(log n) | Sorted list, large data |

* **Importance of Optimizing Search**
* In real e-commerce, thousands of products are searched daily.
* A slow search leads to poor user experience.
* So, binary search is preferred where possible for fast performance.

### ****Analysis****

* **Linear Search** is simple but slow for large lists.
* **Binary Search** is faster (O(log n)), but only works if the data is sorted.
* **Use Binary Search** if the product list is large and sorted.