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

Big O Notation is used to describe the upper bound of an algorithm's running time, i.e., the worst-case scenario. It provides a high-level understanding of the algorithm's efficiency and scalability as the input size grows.

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

* Best Case: The minimum time required for the algorithm to complete, often considering the most favourable conditions.
* Average Case**:** The expected time required for the algorithm to complete, taken as an average of all possible inputs.
* Worst Case: The maximum time required for the algorithm to complete, considering the least favourable conditions.

Compare the time complexity of linear and binary search algorithms.

Linear Search: O(n) in the worst and average cases, where n is the number of products. This is because it may need to check every product.

Binary Search: O(log n) in the worst and average cases, but the products must be sorted. This is because each comparison cuts the search space in half.

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

We use linear search for small datasets or when the data is unsorted and no sorting is feasible.

Binary search is more efficient for large datasets but requires the data to be sorted. It is the better choice if the data remains relatively static (few updates) and you can afford the initial sorting cost.