**Exercise 2: E-commerce Platform Search Function**

An instrument for calculating algorithm efficiency based on input size is Big O notation. It classifies cases into three categories: worst-case (slowest), average-case (typical), and best-case (fastest), which aids in evaluating search performance. In order to choose the best search algorithm for e-commerce platforms and guarantee quick and effective results, it is essential to comprehend these circumstances using Big O notation, particularly when working with big product libraries.

Best-case scenario: The Program would take O(1) to find the item in an unsorted list.

The algorithm's typical performance is shown by the "average-case" scenario. It is O(n/2) for linear search, or just O(n).

The least efficient result is the worst-case situation. This leads to O(n) complexity in linear search when the target element is at the end of the list.

The temporal complexity of linear search is O(n), which means that as the number of entries in the list increases, so does the time required to find an entry. It is hence ineffective for big datasets.

Binary search has an O(log n) time complexity, which makes it much faster. It operates by halving the search interval several times, which makes it incredibly effective for big sorted datasets. Although sorting the data is necessary, the improvement in search time frequently surpasses the sorting overhead.

In conclusion, because binary search has a better time complexity than other methods, it is the recommended choice for huge datasets.