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

**Steps:**

**Understand Asymptotic Notation:**

Big O Notation: Big O notation is a mathematical notation used to describe the upper bound of an algorithm's runtime in terms of the input size. It provides a way to describe how the runtime of an algorithm grows as the size of the input increases. Big O notation helps in analyzing and comparing the efficiency of different algorithms, especially for large inputs.

O(1): Constant time. The runtime does not change with the size of the input.

O(n): Linear time. The runtime grows linearly with the size of the input.

O(log n): Logarithmic time. The runtime grows logarithmically with the size of the input.

O(n^2): Quadratic time. The runtime grows quadratically with the size of the input.

**Best, Average, and Worst-case Scenarios:**

Best-case: The scenario where the algorithm performs the minimum number of operations.

Average-case: The scenario where the algorithm performs an average number of operations over all possible inputs.

Worst-case: The scenario where the algorithm performs the maximum number of operations

**Analysis:**

**Time Complexity:**

**Linear Search:**

Best-case: O(1) (The item is the first element in the array)

Average-case: O(n)

Worst-case: O(n) (The item is not in the array or is the last element)

**Binary Search:**

Best-case: O(1) (The item is the middle element)

Average-case: O(log n)

Worst-case: O(log n) (The item is not in the array or requires multiple comparisons)

**Suitability:**

Linear Search is suitable for small arrays or unsorted arrays where sorting is not feasible due to constraints.

Binary Search is more efficient for large, sorted arrays and is generally preferable when fast search performance is needed and the array is sorted.