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

**Big O Notation**

Big O notation is a measure of the complexity of an algorithm, which is defined as the upper bound of the growth rate of the algorithm's running time or memory usage as the input size increases. Informally, it's a way to describe how fast an algorithm is. Big O is usually expressed as a function of the input size, typically represented as 'n'.

For example, if an algorithm has a time complexity of O(n), it means that the running time of the algorithm grows linearly with the size of the input.

**Best, Average, and Worst-Case Scenarios**

When analyzing algorithms, we consider three cases:

* **Best-case scenario**: The algorithm performs optimally, and the running time is the shortest possible.
* **Average-case scenario**: The algorithm performs reasonably well, and the running time is average.
* **Worst-case scenario**: The algorithm performs poorly, and the running time is the longest possible.

**Implementation:**

Please refer the code.

**Analysis**

Time Complexity Comparison

| **Algorithm** | **Best-Case** | **Average-Case** | **Worst-Case** |
| --- | --- | --- | --- |
| **Linear Search** | **O(1)** | **O(n)** | **O(n)** |
| **Binary Search** | **O(1)** | **O(log n)** | **O(log n)** |