Step 1: Understand Asymptotic Notation

Requirement:

- Explain Big O notation and how it helps in analyzing algorithms.
- Describe the best, average, and worst-case scenarios for search operations.

Implementation:

- **Big O Notation**: The response explains that Big O notation describes the upper bound of an algorithm's running time or space requirements as a function of input size (n), focusing on worst-case performance and ignoring constants/lower-order terms. It's used to compare algorithm efficiency.
- **Search Scenarios**:
  - **Linear Search**:
    - Best Case: O(1) (target at first index).
    - Average Case: O(n) (checks ~n/2 elements on average).
    - Worst Case: O(n) (target at end or not present).
  - **Binary Search**:
    - Best Case: O(1) (target at middle on first check).
    - Average Case: O(log n) (search space halved each step).
    - Worst Case: O(log n) (target at end or not present).