**1. Linear Search**

**Description:**  
• Checks each book record one by one.  
• Works on both **sorted and unsorted** data.  
• Simple to implement and understand.

**Algorithm:**

1. Start from the first book in the list.
2. Compare the title/author with the target.
3. If a match is found, return the index or details.
4. If the end of the list is reached without a match, return "Not Found."

**Time Complexity:**

| **Case** | **Complexity** |
| --- | --- |
| Best | O(1) (match at start) |
| Average | O(n) |
| Worst | O(n) |

**2. Binary Search**

**Description:**  
• Works **only on sorted** data (e.g., books sorted by title or author).  
• Efficiently narrows down the search range by dividing the list in half.

**Algorithm (in words):**

1. Set low = 0 and high = n - 1.
2. Find the middle index.
3. Compare the middle book's title/author with the target.
4. If match found, return index.
5. If target < mid, search left half.
6. If target > mid, search right half.
7. Repeat until match is found or range is empty.

**Time Complexity:**

| **Case** | **Complexity** |
| --- | --- |
| Best | O(1) (match at middle) |
| Average | O(log n) |
| Worst | O(log n) |

**3. Comparison**

| **Feature** | **Linear Search** | **Binary Search** |
| --- | --- | --- |
| Data Requirement | Unsorted / Sorted | Sorted only |
| Time Complexity | O(n) | O(log n) |
| Simplicity | Very simple | Slightly complex |
| Use Case | Small or unsorted data | Large sorted datasets |

**4. When to Use Which**

| **Situation** | **Recommended Search** |
| --- | --- |
| Book data is unsorted | Linear Search |
| Small collection of books | Linear Search |
| Large collection sorted by title | Binary Search |
| Frequent searches required | Binary Search (after sorting once) |