**6\_Library Management System**

**a. Understand Search Algorithms**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Algorithm** | **Description** | **Best Case** | **Average Case** | **Worst Case** | **Sorted Data Required?** |
| **Linear Search** | Scans each element one by one until a match is found. | O(1) | O(n) | O(n) | No |
| **Binary Search** | Repeatedly divides the sorted array in half to find the target. | O(1) | O(log n) | O(log n) | Yes |

**b. Time Complexity**

|  |  |  |  |
| --- | --- | --- | --- |
| 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) |

**c. When to use**

* If your list isn’t sorted, go with linear search, it’s simple, just checks each item one by one. Binary search won’t work unless the list is already sorted.
* If your list is sorted, binary search shines. It cuts the list in half again and again, quickly finding what you’re looking for.
* For small datasets, linear search is often enough, it’s easy to implement and you won’t notice a speed difference.
* But for big datasets, linear search becomes slow. That’s where binary search is way faster, it narrows things down fast instead of checking each item.