WEEK\_1\_ALGORITHMS\_DATA\_STRUCTURES

Exercise 6: **Library Management System**

1. Explain linear search and binary search algorithms.

Linear Search: A search algorithm that sequentially checks each element of a list until the target element is found or the list ends. It has a time complexity of O(n).

Binary Search: A search algorithm that repeatedly divides a sorted list in half, comparing the middle element with the target value. If the target is less than the middle element, it searches the left half; otherwise, it searches the right half. It has a time complexity of O(log n). It requires the entire list to be sorted beforehand.

1. Compare the time complexity of linear and binary search.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Best case | Average case | Worst case |
| Linear Search | O(1) | O(n) | O(n) |
| Binary Search | O(1) | O(log n) | O(log n) |

1. Discuss when to use each algorithm based on the data set size and order.

Linear Search:

* We can use this algorithm when the list is unsorted and the dataset is small.
* We can use this algorithm when search operations are infrequent compared to the cost of maintaining a sorted list, so we have to avoid sorting the entire list again and again.

Binary Search:

* We can use this algorithm when the list is sorted or when multiple search operations justify the cost of sorting.
* We can use this algorithm for large datasets where the efficiency gain from O(log n) time complexity outweighs the cost of sorting.