**Exercise 6: Library Management System**

**Scenario:**  
You are developing a library management system where users can search for books by title or author.

**1. Understand Search Algorithms**

**Linear Search:**

* It checks every element in the array one by one until the target is found or the list ends.
* Simple to implement and doesn’t require the array to be sorted.
* Time Complexity:
  + Best Case: O(1)
  + Average Case: O(n)
  + Worst Case: O(n)

**Binary Search:**

* It works by repeatedly dividing the sorted array in half to search for a value.
* Efficient on large sorted arrays.
* Time Complexity:
  + Best Case: O(1)
  + Average and Worst Case: O(log n)

**2. Setup**

class Book {

int bookId;

String title;

String author;

public Book(int bookId, String title, String author) {

this.bookId = bookId;

this.title = title;

this.author = author;

}

@Override

public String toString() {

return "Book ID: " + bookId + ", Title: " + title + ", Author: " + author;

}

}

**3. Implementation**

import java.util.\*;

public class LibrarySearch {

// Linear Search by Title

public static Book linearSearchByTitle(Book[] books, String title) {

for (Book book : books) {

if (book.title.equalsIgnoreCase(title)) {

return book;

}

}

return null;

}

// Binary Search by Title

public static Book binarySearchByTitle(Book[] books, String title) {

Arrays.sort(books, Comparator.comparing(b -> b.title));

int left = 0, right = books.length - 1;

while (left <= right) {

int mid = (left + right) / 2;

int cmp = books[mid].title.compareToIgnoreCase(title);

if (cmp == 0) return books[mid];

else if (cmp < 0) left = mid + 1;

else right = mid - 1;

}

return null;

}

public static void main(String[] args) {

Book[] books = {

new Book(1, "Java Basics", "John Doe"),

new Book(2, "Data Structures", "Jane Smith"),

new Book(3, "Algorithms", "Alan Turing"),

new Book(4, "Operating Systems", "Andrew Tanenbaum")

};

String searchTitle = "Data Structures";

Book result = linearSearchByTitle(books, searchTitle);

System.out.println("Linear Search Result: " + (result != null ? result : "Not Found"));

result = binarySearchByTitle(books, searchTitle);

System.out.println("Binary Search Result: " + (result != null ? result : "Not Found"));

}

}

Output:

Linear Search Result: [2] Data Structures by Jane Smith

Binary Search Result: [2] Data Structures by Jane Smith

**4. Analysis**

**Time Complexity Comparison:**

* Linear Search: O(n)
* Binary Search: O(log n)

**When to Use:**

* **Linear Search:** Small or unsorted datasets.
* **Binary Search:** Large and sorted datasets.

Binary Search is faster but requires sorted data. Linear Search works without sorting but becomes inefficient as data size increases.