**EXERCISE 6:**

**LIBRARY MANAGEMENT SYSTEM**

**Linear Search:**

A simple search algorithm that checks each element in the array or list sequentially until the target element is found or the list ends.

**Time Complexity:**

Best Case: O(1)

Worst Case: O(n)

**Binary Search:**

A more efficient search algorithm that works on sorted arrays or lists. It repeatedly divides the search interval in half and compares the target value with the middle element.

**Time Complexity:**

Best Case: O(1)

Worst Case: O(log n)

**Usage of search:**

**Linear Search:**

* Suitable for small or unsorted datasets.
* Easier to implement when the list is not sorted.

**Binary Search:**

* Suitable for large, sorted datasets.
* More efficient due to its logarithmic time complexity, but requires the data to be sorted.

class Book {

private int bookId;

private String title;

private String author;

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

this.bookId = bookId;

this.title = title;

this.author = author;

}

public int getBookId() { return bookId; }

public String getTitle() { return title; }

public String getAuthor() { return author; }

@Override

public String toString() {

return "Book{" +

"bookId=" + bookId +

", title='" + title + '\'' +

", author='" + author + '\'' +

'}';

}

}

class Library {

// Linear Search

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

for (int i = 0; i < books.length; i++) {

if (books[i].getTitle().equalsIgnoreCase(title)) {

return i;

}

}

return -1;

}

// Binary Search (assumes the array is sorted by title)

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

int left = 0;

int right = books.length - 1;

while (left <= right) {

int mid = left + (right - left) / 2;

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

if (comparison == 0) {

return mid;

} else if (comparison < 0) {

left = mid + 1;

} else {

right = mid - 1;

}

}

return -1;

}

}

public class LibraryManagementSystem {

public static void main(String[] args) {

Book[] books = {

new Book(1, "The Great Gatsby", "F. Scott Fitzgerald"),

new Book(2, "Moby Dick", "Herman Melville"),

new Book(3, "1984", "George Orwell"),

new Book(4, "To Kill a Mockingbird", "Harper Lee")

};

// Linear Search

String searchTitle = "Moby Dick";

int index = Library.linearSearch(books, searchTitle);

if (index != -1) {

System.out.println("Book found with Linear Search: " + books[index]);

} else {

System.out.println("Book not found with Linear Search.");

}

// Sort the array for Binary Search

java.util.Arrays.sort(books, (b1, b2) -> b1.getTitle().compareToIgnoreCase(b2.getTitle()));

// Binary Search

index = Library.binarySearch(books, searchTitle);

if (index != -1) {

System.out.println("Book found with Binary Search: " + books[index]);

} else {

System.out.println("Book not found with Binary Search.");

}

}

}