**Exercise 6: Library Management System**

**1. Understand Search Algorithms**

**Linear search and binary search algorithms:**

* **Linear Search:** O(n) - Simple, works on unsorted data.
* **Binary Search:** O(log n) - Requires sorted data, more efficient.

**2. Setup**

**Create a Book class:**

public 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;

}

// Getters and Setters

public int getBookId() {

return bookId;

}

public void setBookId(int bookId) {

this.bookId = bookId;

}

public String getTitle() {

return title;

}

public void setTitle(String title) {

this.title = title;

}

public String getAuthor() {

return author;

}

public void setAuthor(String author) {

this.author = author;

}

}

**3. Implementation**

**Implement linear and binary search:**

import java.util.Arrays;

public class BookSearch {

// Linear search

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

for (Book book : books) {

if (book.getTitle().equals(title)) {

return book;

}

}

return null;

}

// Binary search (Assuming array is sorted by title)

public static Book 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().compareTo(title);

if (comparison == 0) {

return books[mid];

} else if (comparison < 0) {

left = mid + 1;

} else {

right = mid - 1;

}

} return null;

}

public static void main(String[] args) {

Book[] books = {

new Book(1, "1984", "George Orwell"),

new Book(2, "To Kill a Mockingbird", "Harper Lee"),

new Book(3, "The Great Gatsby", "F. Scott Fitzgerald"),

new Book(4, "Moby Dick", "Herman Melville")

};

// Linear Search

System.out.println("Linear Search: " + linearSearch(books, "The Great Gatsby"));

// Sorting books by title for binary search

Arrays.sort(books, (b1, b2) -> b1.getTitle().compareTo(b2.getTitle()));

// Binary Search

System.out.println("Binary Search: " + binarySearch(books, "The Great Gatsby"));

}

}

**4. Analysis**

**Time complexity comparison:**

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

**When to use each algorithm:**

* **Linear Search:** Use when the dataset is small or unsorted.
* **Binary Search:** Use when the dataset is large and sorted, offering more efficient search times.