**Exercise 6: Library Management System**

**Understanding the Problem:**  
In a library, users often search for books by title or author. Depending on how books are stored (sorted or not), different search methods offer different performance. Efficient search improves user satisfaction.

**Setup and Implementation:**  
A Book class is created with:

* bookId (int)
* title (String)
* author (String)

Books are stored in an array. We apply both linear search (for unsorted data) and binary search (for sorted data by title).

**Java Code:**

import java.util.\*;

class Book {

int bookId;

String title;

String author;

Book(int id, String t, String a) {

bookId = id;

title = t;

author = a;

}

}

public class LibrarySearch {

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

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

if (books[i].title.equalsIgnoreCase(key)) return i;

}

return -1;

}

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

int l = 0, r = books.length - 1;

while (l <= r) {

int m = (l + r) / 2;

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

if (cmp == 0) return m;

if (cmp < 0) l = m + 1;

else r = m - 1;

}

return -1;

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("How many books? ");

int n = sc.nextInt();

Book[] books = new Book[n];

for (int i = 0; i < n; i++) {

System.out.println("Enter details for book " + (i + 1));

System.out.print("Book ID: ");

int id = sc.nextInt();

System.out.print("Title: ");

String title = sc.next();

System.out.print("Author: ");

String author = sc.next();

books[i] = new Book(id, title, author);

}

System.out.print("Enter title to search: ");

String search = sc.next();

int linear = linearSearch(books, search);

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

int binary = binarySearch(books, search);

System.out.println("Linear Search Index: " + linear);

System.out.println("Binary Search Index: " + binary);

}

}

**Time Complexity Analysis:**

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

**Optimization Discussion:**  
Linear search is simple but slow for large data. Binary search is fast but requires sorted input. For maximum efficiency, using a HashMap<String, Book> where the title is the key provides O(1) average lookup time — making it the best for real-time search in large libraries.