**Exercise 6: Library Management System**

**Q) Explain linear search and binary search algorithms.**

Linear search is a simple searching technique that checks each element in a list one by one until the desired item is found or the end of the list is reached. It does not require the data to be sorted and works well with small or unsorted datasets.

Binary search, on the other hand, is a more efficient algorithm that works only on sorted lists. It repeatedly divides the list in half, comparing the middle element with the target, and narrowing the search range until the target is found or the range becomes empty.

**Code:**

**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;  
 }  
  
 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;  
 }  
  
 public void display(){  
 System.*out*.println("Book ID: "+bookId+", Title: "+title+" , Author: "+author);  
 }  
}

**Library Class: -**

import java.util.Arrays;  
  
public class Library {  
 public void linearSearchByTitle(Book[] books,String title){  
 for(Book b:books){  
 if(b.getTitle().equalsIgnoreCase(title)){  
 b.display();  
 return;  
 }  
 }  
 System.*out*.println("No book found with title: " + title);  
 }  
  
 public void binarySearchByTitle(Book[] books,String title){  
 int left=0,right=books.length-1;  
 while(left<=right){  
 int mid=(left+right)/2;  
 if(books[mid]==null){  
 break;  
 }  
 int cmp=books[mid].getTitle().compareToIgnoreCase(title);  
 if(cmp==0){  
 books[mid].display();  
 return;  
 }else if(cmp<0){  
 left=mid+1;  
 }else{  
 right=mid-1;  
 }  
 }  
 System.*out*.println("No book found with title: " + title);  
 }  
 public void sortBooksByTitle(Book[] books){  
 Arrays.*sort*(books,(b1, b2)->{  
 if(b1==null && b2==null) return 0;  
 if(b1==null) return -1;  
 if(b2==null) return -1;  
 return b1.getTitle().compareToIgnoreCase(b2.getTitle());  
 });  
 }  
  
}

**Main Class: -**

public class Main {  
 public static void main(String[] args) {  
 Book[] books = new Book[5];  
 books[0] = new Book(1,"Java Basics","James Gosling");  
 books[1] = new Book(2,"Python Programming","Guido van Rossum");  
 books[2] = new Book(3,"Data Structures","Mark Allen Weiss");  
 books[3] = new Book(4,"C Programming","Dennis Ritchie");  
 books[4] = new Book(5,"Algorithms","Robert Sedgewick");  
  
 Library library = new Library();  
  
 System.*out*.println("<----Linear Search: Searching for 'Data Structures'---->");  
 library.linearSearchByTitle(books, "Data Structures");  
  
 System.*out*.println("\nSorting books for binary search");  
 library.sortBooksByTitle(books);  
  
 System.*out*.println("\n<----Binary Search: Searching for 'Java Basics'---->");  
 library.binarySearchByTitle(books, "Java Basics");  
 }  
}

**Output:**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**Q) Compare the time complexity of linear and binary search.**

Linear search has a time complexity of O(n), where ‘n’ is the number of elements in the list. This means the search time increases linearly with the size of the list. Binary search has a time complexity of O(log n), making it significantly faster on large, sorted datasets, as it reduces the search space by half in each step.

**Q)** **Discuss when to use each algorithm based on the data set size and order.**

Linear search is ideal for small datasets or when the data is unsorted, as it does not require any preconditions. Binary search should be used when dealing with large datasets that are already sorted, as it offers much better performance in such scenarios.