**Library Management System**

**Step 1: Understand Search Algorithm**

**Linear Search:**

* It checks each item one by one until it finds the targeted element.
* Even if the list is sorted or not it works in the same way.
* It is very simple to write, but can be slow if the data is large.
* Time Complexity:
* Worst case: O(n)

**Binary Search:**

* It only works if the list is already sorted.
* It divides the list into half until it finds the targeted element.
* It is faster than linear search for large elements.
* Time Complexity:
* Worst case: O(log n)

**Step 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;

}

public void display() {

System.out.println("ID: " + bookId + ", Title: " + title + ", Author: " + author);

}

}

**Step 3: Implemetation**

**Linear Search**

public class LinearSearchLibrary {

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

        boolean found = false;

        for (Book book : books) {

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

                book.display();

                found = true;

            }

        }

        if (!found) {

            System.out.println("Book not found using linear search.");

        }

    }

    public static void main(String[] args) {

        Book[] books = {

            new Book(1, "Verity", "Colleen Hoover"),

            new Book(2, "Wings of Fire", "A.P.J. Abdul Kalam"),

            new Book(3, "It Ends With Us", "Colleen Hoover"),

            new Book(4, "It Starts With Us", "Colleen Hoover")

        };

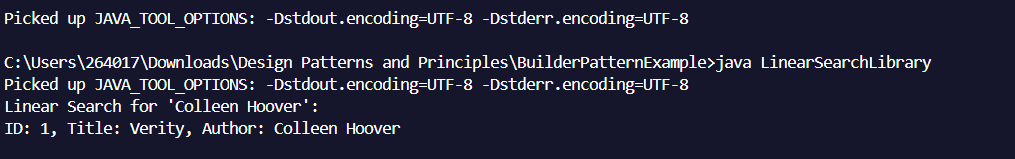
        System.out.println("Linear Search for 'Colleen Hoover':");

        linearSearch(books, "Verity");

    }

}

**Output:**

****

**Binary Search**

public class BinarySearchLibrary {

    public static void binarySearch(Book[] books, String targetTitle) {

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

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

        while (left <= right) {

            int mid = (left + right) / 2;

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

            if (comparison == 0) {

                books[mid].display();

                return;

            } else if (comparison < 0) {

                left = mid + 1;

            } else {

                right = mid - 1;

            }

        }

        System.out.println("Book not found using binary search.");

    }

    public static void main(String[] args) {

        Book[] books = {

            new Book(1, "Verity", "Colleen Hoover"),

            new Book(2, "Wings of Fire", "A.P.J. Abdul Kalam"),

            new Book(3, "It Ends With Us", "Colleen Hoover"),

            new Book(4, "It Starts With Us", "Colleen Hoover")

        };

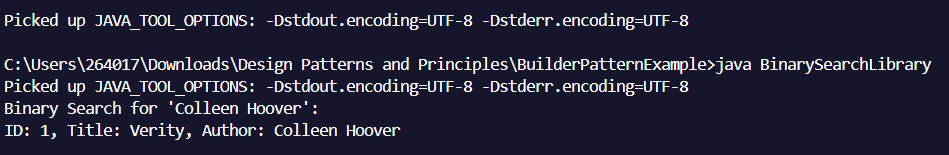
        System.out.println("Binary Search for 'Colleen Hoover':");

        binarySearch(books, "Verity");

    }

}

**Output:**

****

**Step 4: Analysis**

| Algorithm | Time Complexity | When to Use |
| --- | --- | --- |
| Linear Search | O(n) | When data is unsorted or small |
| Binary Search | O(log n) | When data is sorted and large |

**When to use :**

* Linear Search: Use when the book list is small or unsorted.
* Binary Search: Use when the book list is large and sorted by title.