DSA

Module 13

Searching

Author
Srinivas Dande

13. Seraching

- There are two Algorithms for searching an element
  - ✓ Linear Search
  - ✓ Binary Search

Linear Search

- Linear search is a sequential searching algorithm where we start from one end and check every element of the list until the required element is found.
- If the required element is found then we return the index of that element otherwise we return -1.
- It is the simplest searching algorithm.

Binary Search

- Binary Search can be implemented only on a Sorted list of items.
- If the elements are not sorted already, then we need to sort them first.
- In Binary Search, Element is always searched in the middle of a portion of an array and repeatedly dividing the search interval in half.
- If the required element is found then we return the index of that element otherwise we return -1.

13.1. Time Complexity

Operation	Using Iteration	Using Recursion
Linear Search	O(n)	O(n)
Binary Search	O(Log n)	O(Log n)

13.2.1. Linear Search using Iteration

Lab1.java

```
package com.jlcindia.searching;

/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */

public class Lab1 {
	public static int linearSearch(int arr[],int element) {

		for(int i=0;i<arr.length;i++) {
			if(element == arr[i]){
				return i;
			}
		}
		return -1;
	}

	public static void main(String[] args) {

		int arr[] = {10,20,30,40,50,60,70};
		int element = 50;

		int result = linearSearch(arr,element);
		System.out.println(result);
	}
}
```

13.2.2. Linear Search using Recursion

Lab2.java

```
package com.jlcindia.searching;

/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */

public class Lab2 {

	public static int linearSearch(int arr[],int element,int size) {

		if(size==0)
			return -1;
		else if(arr[size-1]==element)
			return size-1;

		return linearSearch(arr, element, size-1);
	}

	public static void main(String[] args) {

		int arr[] = {10,20,30,40,50,60,70};
		int element = 50;
		int size=arr.length;

		int result = linearSearch(arr,element,size);
		System.out.println(result);
	}
}
```

13.3.1. Binary Search using Iteration

Lab3.java

```
package com.jlcindia.searching;
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */
public class Lab3 {
	public static int binarySearch(int arr[],int element) {

		int low=0;
		int high=arr.length-1;

		while(low<=high) {
			int mid = (low+high)/2;

			if(element == arr[mid])
				return mid;
			else if(element > arr[mid])
				low = mid+1;
			else  if(element < arr[mid])
					high = mid-1;
		}

	return -1;
	}
	public static void main(String[] args) {
		int arr[] = {10,20,30,40,50,60,70};
		int element = 25;

		int result = binarySearch(arr,element);
		System.out.println(result);
	}
}
```

13.3.2. Binary Search using Recursion

Lab4.java

```
package com.jlcindia.searching;
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */
public class Lab4 {
	public static int binarySearch(int arr[],int element,int low,int high) {
		if(low>high)
			return -1;

		int mid = (low+high)/2;
			if(element == arr[mid]) {
				return mid;
			}
			else if(element > arr[mid]) {
				low = mid+1;
				return binarySearch(arr,element,low,high);
			}
			else  if(element < arr[mid]) {
					high = mid-1;
					return binarySearch(arr,element,low,high);
			}
			return -1;
	}
	public static void main(String[] args) {
		int arr[] = {10,20,30,40,50,60,70};
		int element = 10;
		int low = 0;		int high=arr.length-1;

		int result = binarySearch(arr,element,low,high);
		System.out.println(result);
	}
}
```

13.4. First Occurence of element in Sorted Array

Lab5A.java

```
package com.jlcindia.searching;

/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */

public class Lab5A {

	public static int firstOccurence(int arr[],int element) {

		for(int i=0;i<arr.length;i++) {
			if(element == arr[i]){
				return i;
			}
		}

	return -1;
	}

	public static void main(String[] args) {

		int arr[] = {10,20,20,30,30,30,50};
		int element = 20;

		int result = firstOccurence(arr,element);
		System.out.println(result);

	}
}
```

13.4. First Occurence of element in Sorted Array

Lab5.java

```
package com.jlcindia.searching;

/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */

public class Lab5 {

	public static int firstOccurence(int arr[], int element) {

		int low = 0;
		int high = arr.length - 1;

		while (low <= high) {
			int mid = (low + high) / 2;

			if (element > arr[mid])
				low = mid + 1;
			else if (element < arr[mid])
				high = mid - 1;
			else if(element==arr[mid]) {
				if(mid==0 || arr[mid-1]!=arr[mid])
					return mid;
				else {
					high= mid-1;
				}
			}
		}

		return -1;
	}
```

```
        public static void main(String[] args) {
                int arr[] = { 10, 20, 20, 30, 30, 30, 50 };
                int element = 20;
                int result = firstOccurence(arr, element);
                System.out.println(result);
        }
}
```

13.5. Last Occurence of element in Sorted Array

Lab6A.java

```
package com.jlcindia.searching;
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */
public class Lab6A {
        public static int lastOccurence(int arr[],int element) {
                for(int i=arr.length-1;i>=0;i--) {
                        if(element == arr[i]){
                                return i;
                        }
                }
        return -1;
        }
        public static void main(String[] args) {
                int arr[] = {10,20,20,30,30,30,50};
                int element = 30;
                int result = lastOccurence(arr,element);
                System.out.println(result);
        }
}
```

13.5. Last Occurence of element in Sorted Array

Lab6.java

```
package com.jlcindia.searching;
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */
public class Lab6 {
	public static int lastOccurence(int arr[], int element) {
		int low = 0;
		int high = arr.length - 1;

		while (low <= high) {
			int mid = (low + high) / 2;

			if (element > arr[mid])
				low = mid + 1;
			else if (element < arr[mid])
				high = mid - 1;
			else if(element==arr[mid]) {
				if(mid==arr.length - 1 || arr[mid]!=arr[mid+1])
					return mid;
				else {
					low= mid+1;
				}
			}
		}
		return -1;
	}
	public static void main(String[] args) {
		int arr[] = { 10, 20, 20, 30, 30, 30, 50 };
		int element = 30;
		int result = lastOccurence(arr, element);
		System.out.println(result);
	}
}
```

13.6. Count Occurence of element in Sorted Array

Lab7.java

```
package com.jlcindia.searching;

/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */

public class Lab7 {

	public static int firstOccurence(int arr[], int element) {

		int low = 0;
		int high = arr.length - 1;

		while (low <= high) {
			int mid = (low + high) / 2;

			if (element > arr[mid])
				low = mid + 1;
			else if (element < arr[mid])
				high = mid - 1;
			else if (element == arr[mid]) {
				if (mid == 0 || arr[mid - 1] != arr[mid])
					return mid;
				else {
					high = mid - 1;
				}
			}
		}

		return -1;
	}
```

```
	public static int lastOccurence(int arr[], int element) {
		int low = 0;
		int high = arr.length - 1;
		while (low <= high) {
			int mid = (low + high) / 2;

			if (element > arr[mid])
				low = mid + 1;
			else if (element < arr[mid])
				high = mid - 1;
			else if (element == arr[mid]) {
				if (mid == arr.length - 1 || arr[mid] != arr[mid + 1])
					return mid;
				else {
					low = mid + 1;
				}
			}
		}
		return -1;
	}
	public static int countOccurence(int arr[], int element) {
		int first = firstOccurence(arr, element);
		if (first == -1)
			return 0;

		int last = lastOccurence(arr, element);
		int count = last - first + 1;
		return count;
	}
	public static void main(String[] args) {
		int arr[] = { 10, 20, 20, 30, 30, 30, 50 };
		int element = 15;
		int result = countOccurence(arr, element);
		System.out.println(result);
	}
}
```