practical 1

Name: Shantanu Sethi

Roll No: 163

Subject: DS Lab using JAVA

Semester: I (AY 2024-26)

Batch: C

Aim: To implement searching techniques using JAVA

1. Linear search
2. Binary Search
3. Recursive binary search

Objectives:

* 1. To know what searching is and understand algorithm for Linear search and Binary search
  2. To understand the basic principles of recursive definitions and functions and be able to write simple recursive function

Theory:

Searching: It is process of checking and finding an element from list of elements.

* 1. Linear search
  2. Binary search

Pseudocode for Linear search:

Algorithm linear (a, n, key)

// key is data to be searched in array a of size length

Pre: Unsorted list of length n.

Post: If found, return position of key in array a. If key not present in list, return negative value

* 1. for i = 0 to (n - 1) do

if (key == a[i]) return i

* 1. return -1

Pseudocode for Binary search:

Algorithm binary\_search (a, n, key) // key - data to be searched in array a of size n

Pre: Sorted list of length n.

Post: If present, return position of key in array a; Else return -1

* 1. low =0
  2. high = n-1
  3. while (low <= high)

1. mid = (low + high)/2 2. if (key == a[mid])

return mid

3. if ( key < a[mid]) high = mid -1

4 else low = mid + 1

1. return -1

Pseudocode for Recursive binary search:

BinarySearch(a, key, low, high)

1. if (low > high)

return -1 // not found

1. mid = (low + high) / 2
2. if (key < a[mid] ) return BinarySearch ( a, key, low, mid-1)
3. else if (key > a[mid]) return BinarySearch ( a, key, mid+1, high)
4. else

return mid // found

Program:

JAVA implementation for Linear search:

package search;

public class LinearSearch { static String LinearSearch(int[]arr,int key) { for(int i=0;i<arr.length;i++) { if(arr[i]==key) {

return"The element is found at position "+(i+1);

}

}

return"Element not found";

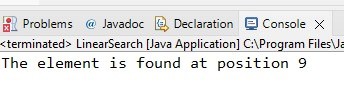
}

public static void main(String[]args) { int[]arr= {10,20,30,40,50,60,70,80,90,100}; System.*out*.println(*LinearSearch*(arr,90));

}

}

Output:



JAVA implementation for Binary search:

packagesearch;

public class BinarySearch { staticint BinarySearch(int[] arr, int key) {

int start=0; int end =arr.length-1;

while(start<=end) { int mid=(start+end)/2; if(arr[mid]==key) {

return mid;

} else if(arr[mid]>key) {

end=mid-1;

} else if(arr[mid]<key) {

start=mid+1;

}

}

return -1;

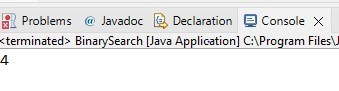
}

public static void main(String[]args) { int[]arr= {0,10,20,30,40,50,60}; System.*out*.println(*BinarySearch*(arr,40));

}

}

Output:



JAVA implementation for Recursive Binary search:

package search;

public classrecursiveBinarySearch {

static int recursiveBinarySearch(int[]arr,int key,int start,int end) { if(start>end) { return-1;

}

int mid=(start+end)/2; if(key<arr[mid]) { return *recursiveBinarySearch*(arr,key,start,mid-1);

}

else if(key>arr[mid]) { return *recursiveBinarySearch*(arr,key,mid+1,end);

}

else { return mid;

}

}

main(String[] args) { public static void int[]arr= {0,0,20,30,40,50,60,70}; int end=arr.length-1;

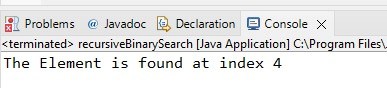
int index=*recursiveBinarySearch*(arr,40,1,end);

System.*out*.println("The Element is found at index "+index);

}

}

Output:



Conclusion:

Linear search is used for an unsorted small list of elements. It has a time complexity of

O(n),

Binary Search is used to search through large sorted arrays. It has a time complexity of

O(log n)