Data Structures and Algo in Java - Day 19

Started a new Concept call Binary Search,

did some basic level problems in BS

public class day19 {

public static void main(String[] args) {

int arr [] = {3,4,6,7,9,12,16,17};

int tar = 8;

floor(arr,tar);

}

public static void getValues1(int arr [] , int target)

{

int low = 0;

int high = arr.length - 1 ;

int ans = binarySearch(arr, target, low, high);

if (ans==-1)

{

System.out.println("Not Found");

}

else{

System.out.println("Found in the index: "+ans);

}

}

public static int binarySearch(int arr[], int tar, int low , int high)

{

if(low>high) return -1;

int mid = low+(high-low)/2;

if(arr[mid]==tar) return mid;

if(arr[mid]<tar) return binarySearch(arr, tar, mid+1, high);

return binarySearch(arr, tar, low, mid-1);

}

public static void lowBound(int arr[],int target)

{

int low = 0;

int high = arr.length-1;

int ans = arr.length;

while(low<=high)

{

int mid = (low+high)/2;

if(arr[mid]>=target)

{

ans = mid;

high = mid -1;

}

else

{

low = mid+1;

}

}

System.out.println("The lower bound is in the index "+ans);

}

public static void upperBound(int arr[],int tar)

{

int low = 0;

int n = arr.length;

int high = n-1;

int ans = n;

while(low<=high)

{

int mid = (low+high)/2;

if(arr[mid]>tar)

{

ans = mid;

high = mid -1;

}

else

{

low = mid +1;

}

}

System.out.println("The Upper bound is in the index "+ans);

}

public static void insertAtCorrectPosition(int arr[],int tar)

{

int low = 0;

int high = arr.length-1;

int ans = arr.length;

while(low<=high)

{

int mid = (low+high)/2;

if(arr[mid]>=tar)

{

ans = mid;

high = mid -1;

}

else

{

low = mid+1;

}

}

System.out.println("The Correct Position to insert the target element is at the index "+ans);

}

public static void ceil(int arr [], int tar)

{

int low = 0;

int high = arr.length-1;

int ans = -1;

while(low<=high)

{

int mid = (low+high)/2;

if(arr[mid]>=tar)

{

ans = arr[mid];

high = mid -1;

}

else

{

low = mid+1;

}

}

if(ans>-1)

{

System.out.println("The Ceil is "+ans);

}

else

{

System.out.println("No Ceil");

}

}

public static void floor(int arr[],int tar)

{

int low = 0;

int high = arr.length -1;

int ans = -1;

while(low<=high)

{

int mid = (low+high)/2;

if(arr[mid]<=tar)

{

ans = arr[mid];

low = mid+1;

}

else

{

high = mid-1;

}

}

if(ans>-1)

{

System.out.println("The floor is "+ans);

}

else

{

System.out.println("No floor");

}

}

}