Data structures and algo in JAVA - Day 21

public class day22

{

public static void main(String[] args)

{

// int arr [] = {1,1,2,2,3,3,4,5,5,6,6};

// int arr[] = {3};

// int arr [] = {1,2,2,3,3,4,4};

// int arr [] = {1,1,2,2,3,3,4};

// int ans = singleElementInSortedArray(arr);

// if(ans==-1)

// {

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

// }

// else{

// System.out.println(ans+" is the single element");

// }

//

// int arr [] = {10,9,8,7,6};

// int arr [] = {1,2,3,4,5};

// int arr [] = {1,2,3,4,5,6,7,8,7,6,5};

int arr [] = {1,5,1,2,1};

int ans = findOnePeakElemenet(arr);

if(ans==-1)

{

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

}

else{

System.out.println(ans+" is the Peak element");

}

}

public static int singleElementInSortedArray(int arr[])

{

int n = arr.length;

if(n==1)

{

return arr[0];

}

if(arr[0]!=arr[1])

{

return arr[0];

}

if(arr[n-1]!= arr[n-2])

{

return arr[n-1];

}

int low = 1;

int high = n-2;

while(low<=high)

{

int mid = (low+high)/2;

if(arr[mid]!=arr[mid+1] && arr[mid]!=arr[mid-1])

{

return arr[mid];

}

if((mid%2==0 && arr[mid]==arr[mid+1])||(mid%2==1 && arr[mid]== arr[mid-1]))

{

low = mid+1;

}

else

{

high = mid-1;

}

}

return -1;

}

public static int findOnePeakElemenet(int arr[])

{

int n = arr.length;

if(n==1)

{

return arr[0];

}

if(arr[0]>arr[1])

{

return arr[0];

}

if(arr[n-1]>arr[n-2])

{

return arr[n-1];

}

int low = 1;

int high = n-2;

while(low<=high)

{

int mid = (low+high)/2;

if(arr[mid]>arr[mid+1] && arr[mid]>arr[mid-1])

{

return arr[mid];// program will always end here

}

else if(arr[mid]>arr[mid-1])

{

low = mid +1;

}

else

{

high = mid -1;

}

}

return -1; /\* program will never come to this point but

the compiler wont accept a return type inside a conditional statement

so we use -1 just because it has a integer return type\*/

}

}