30 Days Leetcode challenge - Day 14

public class leetday14

{

class Solution1 {

public int searchInsert(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;

}

}

return ans;

}

}

class Solution2 {

public int[] searchRange(int[] arr, int x)

{

int low = 0;

int n = arr.length;

int high = n-1;

int lb = lowerBound(arr,x,low,high);

int ub = upperBound(arr, x, low, high);

int ans [];

if (lb == n || arr[lb]!=x)

{

ans = new int []{-1,-1};

return ans;

}

else

{

ans = new int []{lb,ub-1};

return ans;

}

}

public static int lowerBound(int arr[],int target,int low,int high)

{

int lb = arr.length;

while(low<=high)

{

int mid = (low+high)/2;

if(arr[mid]>=target)

{

lb = mid;

high = mid -1;

}

else

{

low = mid+1;

}

}

return lb;

}

public static int upperBound(int arr[],int target,int low,int high)

{

int ub = arr.length;

while(low<=high)

{

int mid = (low+high)/2;

if(arr[mid]>target)

{

ub= mid;

high = mid -1;

}

else

{

low = mid+1;

}

}

return ub;

}

}

}