**Day-25**

**Python DSA**

**Implement Upper Bound**

<https://www.naukri.com/code360/problems/implement-upper-bound_8165383?leftPanelTabValue=PROBLEM>

def upperBound(arr: [int], x: int, n: int) -> int:

low, high =0,n

while low<high:

mid= (low + high)//2

if arr[mid]>x:

high=mid

else:

low= mid+1

return low

Time Complexity – O(log N)

Space Complexity – O(1)

**Leetcode 35 Search Insert Position**

class Solution:

def searchInsert(self, nums: List[int], target: int) -> int:

n= len(nums)

low=0

high=n-1

ans=n

while low<=high:

mid= (low + high)//2

if nums[mid] >= target:

ans= mid

high = mid-1

else:

low= mid+1

return ans

Time Complexity- O(log N)

Space Complexity – O(1)