**Day-33**

**Python DSA**

**Leetcode 1482 Minimum Number of Days to Make m Bouquets**

<https://leetcode.com/problems/minimum-number-of-days-to-make-m-bouquets/description/>

class Solution:

def minDays(self, bloomDay: List[int], m: int, k: int) -> int:

n= len(bloomDay)

if m\*k>n:

return -1

def canMake(days):

bouquets=0

flowers=0

for day in bloomDay:

if day<=days:

flowers+=1

if flowers==k:

bouquets+=1

flowers=0

else:

flowers=0

if bouquets >= m:

return True

return False

l,r= min(bloomDay), max(bloomDay)

res=-1

while l<=r:

mid=(l+r)//2

if canMake(mid):

res=mid

r=mid-1

else:

l=mid+1

return res

TC – O(n \* log(max(bloomDay)))

SC- O(1)

**Leetcode 1011 Capacity To Ship Packages Within D Days**

<https://leetcode.com/problems/capacity-to-ship-packages-within-d-days/description/>

class Solution:

def shipWithinDays(self, weights: List[int], days: int) -> int:

left,right= max(weights),sum(weights)

res=right

while left <= right:

mid= (left + right)//2

day\_count =1

current\_weight =0

for w in weights:

if current\_weight + w> mid:

day\_count +=1

current\_weight=0

current\_weight +=w

if day\_count<=days:

res=mid

right=mid-1

else:

left= mid+1

return res

TC – O(n\* log(sum(weights)))

SC- O(1)