Brute Force

def roseGarden(arr: List[int], r: int, b: int):

    if r\*b>len(arr):#as we have only len(arr) roses

        return -1

    for i in range(min(arr),max(arr)+1):

        count=0

        bo=0

        for j in range(len(arr)):

            if arr[j]<=i:

                count+=1

            else:#goes only when greater element such a case exist where all are less and it went on incrementing

                bo+=(count//r)

                count=0

        bo+=(count//r)  # Account for the last set of roses#thus we need this to check

        if bo >= b:

            return i

    return -1

Optimal:

The min ->max goes in sorted order so binary search possible

def roseGarden(arr: List[int], r: int, b: int):

    if r\*b>len(arr):#as we have only len(arr) roses

        return -1

    def func(arr,i,r,b):

        count=0

        bo=0

        for j in range(len(arr)):

            if arr[j]<=i:

                count+=1

            else:#goes only when greater element such a case exist where all are less and it went on incrementing

                bo+=(count//r)

                count=0

        bo+=(count//r)  # Account for the last set of roses#thus we need this to check

        if bo >= b:

            return i

        else:

            return -1

    s=min(arr)

    e=max(arr)

    mini=float('inf')

    while(s<=e):

        mid=(s+e)//2

        a=func(arr,mid,r,b)

        if a==-1:

            s=mid+1

        else:

            e=mid-1

            mini=min(mid,mini)

    return mini