ASSIGNMENT - 5

ANIKEIT SETHI (190001003)

Binary Search used here to calculate the minimum time it would take for the painters to complete their work. We are able to use **binary Search** here because if time t is sufficient for the workers to complete their task, then any value of time greater than t will also work hence, we can reduce our search space by applying binary search here. For a particular time t we can check that whether the task can be finished by the workers in the given time frame or not in O(n).Hence the time Complexity is **O(nlog(sum of elements in the array))**

Code:-

# We have to paint n boards of length {A1, A2…An}.  
# There are k painters available and each takes 1 unit time to paint 1 unit of board.  
# The problem is to find the minimum time to get this job done under the constraints  
# that any painter will only paint continuous sections of boards, say board {2, 3, 4}  
# or only board {1} or nothing but not board {2, 4, 5}.

def numberOfPainters(arr, n, maxLen):  
 total = 0  
 numPainters = 1  
  
 for i in arr:  
 total += i  
  
 if total > maxLen:  
 # for next count  
 total = i  
 numPainters += 1  
  
 return numPainters  
  
  
def partition(arr, n, k):  
 lo = max(arr)  
 hi = sum(arr)  
  
 while lo < hi:  
 mid = lo + (hi - lo) / 2  
 requiredPainters = numberOfPainters(arr, n, mid)  
  
 if requiredPainters <= k:  
 hi = mid  
  
  
 else:  
 lo = mid + 1  
  
 # required  
 return lo  
  
  
# Driver code  
  
n, k = list(map(int, input().split()))  
arr = list(map(int, input().strip().split()))[:n]  
print(int(partition(arr, n, k)))

Output: -



