Tutorial3

● Graded

Student

Abhinav Shripad

Total Points

3 / 3 pts

Question 1

(no title) **3** / 3 pts

+ 0.6 pts Written "I do not know how to approach this problem" - 0.6 pointsCorrect

✔ **+ 2 pts** Binary search idea + correctness

✔ **+ 1 pt** Showing $O(n \log (\max (L)))$ time complexity

+ 1.25 pts On the right tract with binary search idea but informal/no correctness

+ 0 pts Incorrect

perfect!

COL351: Analysis and Design of Algorithms
Tutorial 3

Name: Abhinav R. Shripad
Entry number: 2022CS11596
Date: August 13, 2024
Group: 3

Algorithm:- from a peice of length e, I can get $\lfloor e/d \rfloor$ peices of length d.
Observe that $\sum_{i=1}^{n} \lfloor \frac{a_i}{d} \rfloor$ where a_i are element of array, this is a "decreasing function". Hence we can apply binary search to find max d such that $\sum_{i=1}^{n} \lfloor \frac{a_i}{d} \rfloor \geq k$.

Code :-

(*) $O(n \log(mx)) = O(n \cdot \text{bits in } mx)$
$= O(\text{total bits in } L) = \text{polynomial time}$

```
mx = maximum(L)  # maximum element in array L.
low:=1, high:= mx , ans =-1, mid = 0;
while (low <= high):
    mid = low + (high-low)/2
    total = sum_{i=1}^{n} floor(L[i]/mid)   # O(n)
    if ( total >= k):
        ans = mid, low=mid + 1
    else:
        high=mid-1
return ans   # ans == -1 -> no solution
```

-> T.C. $= O(n \log(mx)) = O(n \times \text{no. of bits in } mx)$ (*)
$= O(\text{total bits to represent } L)$ -> polynomial