Koko Eating Bananas

Tuesday, 16 November 2021 9:02 PM

1. Koko loves to eat bananas. There are n piles of bananas, the ith pile has piles[i] bananas. The guards have gone and will come back in h hours.
2. Koko can decide her bananas-per-hour eating speed of k. Each hour, she chooses some pile of bananas and eats k bananas from that pile. If the pile has less than k bananas, she eats all of them instead and will not eat any more bananas during this hour.
3. Koko likes to eat slowly but still wants to finish eating all the bananas before the guards return.
4. Return the minimum integer k such that she can eat all the bananas within h hours.

Size of array

time = ? H hr

speed of eating bananas = 6 bananas/hr.

$\left\lceil \frac{pile[i]}{k} \right\rceil$ time required for ith piles

hour = 8 hrs

array → 3 6

↑ no. of bananas in ith piles

ith pile have 11 bananas

↳ Time = $\left\lceil \frac{11}{3} \right\rceil_{ceil} = \lceil 3.6 \rceil_{ceil} = 4$

Maths → ceil for finding time.

first hr - 3
Second hr = 3

7 1 1

Speed of eating banans = 3 ba/hr.

↳ Time?

koko can decide speed of eating bananas per hour,

speed = k bananas/hr

↳ speed

Minimise 'k'

H = 3 hrs

piles → 3 + 6 + 7 + 11   n = 4

H = 15   H = 27

index: 0, 1, 2, 3

H = 8 hrs

Speed of eating bananas = K → minimise 'k'

K = 2   time → 2 + 3 + 4 + 6 = 15 hrs

Min time required to eat complete pile = size of pile is minimum hr required

time = $\left\lceil \frac{piles[i]}{k} \right\rceil$ ceil

Min. speed = (Max from piles) = 11 bananas/hr

K = 4 bananas/hr   time → 1 + 2 + 2 + 3 = 8 hr

K = 3 bananas/hr   time → 1 + 2 + 3 + 4 → 10 hrs   Invalid situation

n → size of piles

0 < K ≤ { max from piles }

hours required ← H < n

piles → 3 6 7 11

0 1 2 3

time$_1$ → 1 + 2 + 2 + (3) = 8 hrs

time$_2$ → 2 + 3 + 4 + 6 = 15 hrs

time$_3$ → 1 + 2 + 3 + 4 = 10 hrs

time$_4$ → 1 + 2 + 2 + 3 = 8 hrs

H = 8 hr

K = ~~8~~ 4

time required for i^{th} pile is $\left\lceil \frac{piles[i]}{k} \right\rceil_{ceil}$

lo = ~~0~~ ~~3~~ 4

hi = ~~11~~ ~~4~~ 3

Running → $lo <= hi$

Stop → $lo > hi$

$speed_1 = \frac{lo+hi}{2} = \frac{11}{2} = (6)$

$speed_2 = \frac{lo+hi}{2} = \frac{4}{2} = (2)$

$speed_3 = \frac{7}{2} = (3)$

$speed_4 = \frac{lo+hi}{2} = \frac{8}{2} = (4)$

piles → 3 6 7 11 2^{32}

$k = 1$ (min)

H = 8 hrs.

first 1 hr → 2, rem = 1

1 hr → 1

i = 2

(2 hrs)

Time = n log(max. value)

2^{32}

32n

Complexity required for binary search

Speed = (k) minimise

Ex → k = 2

k = 6 bananas/hr.

k = 4 bananas/hrs.

invalid condition → H < n

kmax = Max from piles

T_6 = [1 + 1 + 2 + 2] = 6 hrs ✓

T_3 = 1 + 2 + 3 + 4 = 10 hrs

T_4 = 1 + 2 + 2 + 3 = 8 hrs ✓

[1]

Stop condition

lo > hi

lo = 1 4

hi = 11 5 3

Speed (Mid) = $\frac{lo + hi}{2} = \frac{1 + 11}{2} = 6$

$S = \frac{1 + 5}{2} = \frac{6}{2} = 3$

$S = \frac{4 + 5}{2} = 4$

Running → lo <= hi

Find the Smallest Divisor Given a Threshold

Tuesday, 16 November 2021 10:40 PM

threshold = 11

Given an array of integers `nums` and an integer `threshold`, we will choose a positive integer `divisor`, divide all the array by it, and sum the division's result. Find the **smallest** `divisor` such that the result mentioned above is less than or equal to `threshold`. Smallest divisor → Minimise

Each result of the division is rounded to the nearest integer greater than or equal to that element. (For example: `7/3 = 3` and `10/2 = 5`). → ceil

It is guaranteed that there will be an answer.

threshold → min ⇒ length of array.

2 3 5 7 11

↳ When threshold is min.

Value of divisor = (Max. from array) [including minimisation of k]

k = ?

↳ divisor

Sum of result of division ≤ threshold.

Min. possible value of k.

lo = 1

hi = [Max from array]

2 3 5 7 11

threshold = 11

Sum_6 = 1 + 1 + 1 + 2 + 2 = 7

div = 6 ~~∞~~ 2

Sum_8 = 1 + 1 + 2 + 3 + 4 = 11

lo = ~~1~~ 2 3 Sum_1 = 2 + 3 + 5 + 7 + 11 = greater than 11

hi = ~~11~~ 6 2 Sum_2 = 1 + 2 + 3 + 4 + 6 = 11

Mid = $\frac{1+11}{2}$ = 6

Mid = $\frac{1+5}{2}$ = $\frac{6}{2}$ = 3

Mid = $\frac{1+2}{2}$ = 1

Mid = $\frac{2+2}{2}$ = 2

lo > hi] stop

Allocate Minimum Number Of Pages

Tuesday, 16 November 2021 10:55 PM

5 books, 6 students, Distribution?!

N = 4, M → no. of students

Ex → 10 (0) 30 (1) 20 (2) 40 (3)

1. You are given N number of books. Every ith book has Ai number of pages.
2. You have to allocate books to M number of students. There can be many ways or permutations to do so. In each permutation, one of the M students will be allocated the maximum number of pages. Out of all these permutations, the task is to find that particular permutation in which the maximum number of pages allocated to a student is minimum of those in all the other permutations and print this minimum value.
3. Each book will be allocated to exactly one student. Each student has to be allocated at least one book.
4. Note: Return -1 if a valid assignment is not possible, and allotment should be in contiguous order.

books < students

M = 2

D_1 → st(1) 10 → 30+20+40 = 90 (st(2)) → Bunder = 90 Max

D_2 → 10+30 = 40 Page; 20+40 = 60 Pages. Bunder = 60

D_3 → 10+30+20 = 60; 40 = 40

Maximum Bunder

Min → 60

Bunder Minimise

condition → 1 book is allocate to 1 student

st1 st2 st3 st4

10 30 20 ~~40~~

st(1) → 10+30+20+40 = 100 (sum of array)

Distribute

Bunder = 40

Pages → 10 30 20 40

0 1 2 3

M=3

D1 → 10 30 20+40=60 →

D2 → 10 30+20=50 40 →

D3 → 10+30=40 20 40 →

10+30 20 40

Burden (max in individual distribution)

60
50
40

Min → 40

40 → 40

Pages → 10 30 20 40 - - -

M = 2

St 1

Sum = 10 + 30 + 20

St 2 =

Sum = 60

Allocation = 60

lo = 40 ~~50~~

hi = 100 ~~69~~ 61 59 ~~60~~ M

Mid = $\frac{100+40}{2}$ = 70

Mid = $\frac{109}{2}$ = 54

Mid = $\frac{124}{2}$ = 69

St 1

Sum = 10 + 30.

St 2

Sum = 20

St 1

Sum = 10 + 30 + 20 = 60

St 2

Sum = 40

Allocation = 62

Mid = $\frac{116}{2}$ = 58

St 1

Sum = 10 + 30

St 2

Sum = 20

St 3

Sum = 40

St = 1

Sum = 10 + 30 + 20 = 60

St = 2

Sum = 40

Allocation = 70

(Burden)

St 1

Sum = 10 + 30

St

Mid = 59 Sum 20

X

Mid = 60