# **Leetcode 875 – Koko Eating Bananas**

## Problem Understanding

Koko loves to eat bananas. There are piles[] of bananas and she can eat at most k bananas per hour.  
Each hour, she chooses a pile and eats up to k bananas. If the pile has fewer than k bananas, she eats all.

Return the **minimum integer k** such that she can eat all the bananas within h hours.

Example:

Input: piles = [3, 6, 7, 11], h = 8

Output: 4

## Optimized Java Solution

class Solution {

public int minEatingSpeed(int[] piles, int h) {

int n = piles.length;

long total = 0;

for (int p : piles) {

total += p; // total bananas

}

// Lower bound (at least total/h bananas per hour)

int left = (int) ((total - 1) / h) + 1;

// Upper bound (if she eats most evenly)

int right = (int) ((total - n) / (h - n + 1)) + 1;

while (left < right) {

int mid = left + (right - left) / 2;

int time = 0;

for (int m : piles) {

time += (m - 1) / mid + 1; // same as ceil(m / mid)

}

if (time > h) {

left = mid + 1; // not enough speed

} else {

right = mid; // try smaller speed

}

}

return left;

}

}

## Key Observations

* ceil(x / y) = (x - 1) / y + 1 → used to avoid floating point
* Lower bound: minimum average speed = ceil(total / h)
* Upper bound: a tighter value using (total - n) / (h - n + 1) + 1

## Dry Run with Example

piles = [3, 6, 7, 11], h = 8

total = 27

left = (27 - 1)/8 + 1 = 4

right = (27 - 4)/(8 - 3 + 1) + 1 = 5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| left | right | mid | time (hours) | action |
| 4 | 5 | 4 | 8 | right = 4 |
| 4 | 4 | - | - | ✅ Done |

👉 Answer: 4

## Time / Space Complexity

* **Time:** O(n \* log(maxPile))
  + Each binary search iteration: O(n) time to calculate total hours
* **Space:** O(1) – constant extra space

## Alternate Approaches

* Brute Force (Try all k from 1 to max(piles)) → TLE for large inputs
* Use Math.ceil() if allowed (but (x - 1)/k + 1 avoids floating points)

## Similar Problems

* [📌 1011. Capacity To Ship Packages Within D Days](https://leetcode.com/problems/capacity-to-ship-packages-within-d-days/)
* [📌 1482. Minimum Number of Days to Make m Bouquets](https://leetcode.com/problems/minimum-number-of-days-to-make-m-bouquets/)
* [📌 410. Split Array Largest Sum](https://leetcode.com/problems/split-array-largest-sum/)