Koko Eating Bananas

Question:

https://leetcode.com/problems/koko-eating-bananas/

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.

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.

Koko likes to eat slowly but still wants to finish eating all the bananas before the guards' return.

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

Example 1:

`Input: piles = [3,6,7,11], h = 8`
`Output: 4`
Example 2:

`Input: piles = [30,11,23,4,20], h = 5`
`Output: 30`
Example 3:

`Input: piles = [30,11,23,4,20], h = 6`
`Output: 23`

Approach 1:

We will start from 1 and check when each number until we find the correct answer.

It is a simple linear search method or brute force search.

Solution 1:

https://gist.github.com/vermaayush680/d0c72f77189d6014cd76c2fae813257e

```
speed = 1
while True:
        hour_spent = 0
        for pile in piles:
                hour_spent += math.ceil(pile / speed)
        if hour_spent <= h:
                return speed
        else:
                speed += 1
```

Approach 2:

Instead of using brute force, I used Brute force between the constraints to find the optimal solution.

Basically converting Linear Search to Binary Search.

The value of l,r are from constraints.

Solution 2:

https://gist.github.com/vermaayush680/eb7a897b7718f9ff8c0860d5ca6f8c8d

```
l,r=1,10**9
while(l<r):
        m=l+(r-l)//2
        a=0
        for p in piles:
                a+=(p+m-1)//m
        if a>h:
                l=m+1
        else:
                r=m
return l
```