875. Koko Eating Bananas

Created	@October 15, 2022 2:30 PM
Difficulty	Medium
LC Url	https://leetcode.com/problems/koko-eating-bananas/
Importance	
Tag	Binary search NEET
Video	

Koko loves to eat bananas. There are `n` piles of bananas, the `i` `th` 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
```

Constraints:

- `1 <= piles.length <= 10 4`
- `piles.length <= h <= 10 9`
- `1 <= piles[i] <= 10 9`

Solution

二分查找定位速度（最大值最小化问题，Java） - 爱吃香蕉的珂珂 - 力扣（LeetCode）

```
class Solution:
    def minEatingSpeed(self, piles: List[int], h: int) -> int:
        max_val = max(piles)
        left, right = 1, max_val

        while left + 1 < right:
            speed = (left + right) // 2
            if self.total_time(piles, speed) > h:
                left = speed
            else:
                right = speed

        if self.total_time(piles, left) <= h:
            return left

        if self.total_time(piles, right) <= h:
            return right

        return -1

    def total_time(self, piles, speed):
        t = 0
        for pile in piles:
            t += (pile + speed - 1) // speed
        return t
```

复杂度分析：

- 时间复杂度：$O(N \log \max(piles))$，这里 N 表示数组 `piles` 的长度。我们在 $[1, \max piles]$ 里使用二分查找定位最小速度，而每一次执行判别函数的时间复杂度是 $O(N)$；
- 空间复杂度：$O(1)$，算法只使用了常数个临时变量。