

Input: arr[] = {7, 10, 4, 3, 20, 15}, K=3

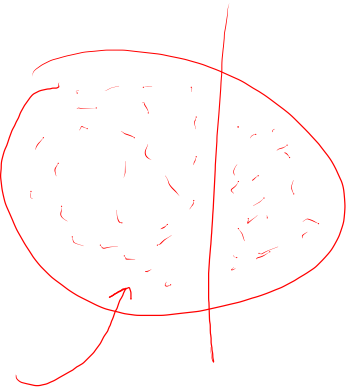
Output: 7 (small)

Input: arr[] = {7, 10, 4, 3, 20, 15}, K=4

Output: 10 (large)

7 | 10 | 4 | 3 | 20 | 15

maxEle

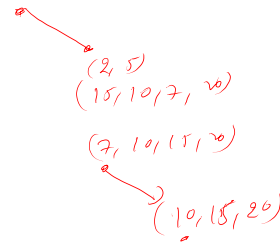


① $\rightarrow 7 \rightarrow 20 \rightarrow 15 \rightarrow 10 \rightarrow 7 \rightarrow 7 \rightarrow 7$
T: ?

② $\text{arr} \cdot \text{sort}()$
 return arr[k-1]

[7, 4, 7, 10, 15, 20] n=6
0 1 2 3 4 5

③ $\begin{bmatrix} 7, 10, 4, 3, 20, 15 \\ 0, 1, 2, 3, 4, 5 \end{bmatrix}$
 $\begin{bmatrix} 3, 4, 15, 10, 7, 20 \\ 0, 1, 2, 3, 4, 5 \end{bmatrix}$



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

1 20

rate = 15

30 = 15 * 2

11 = 15 * 1

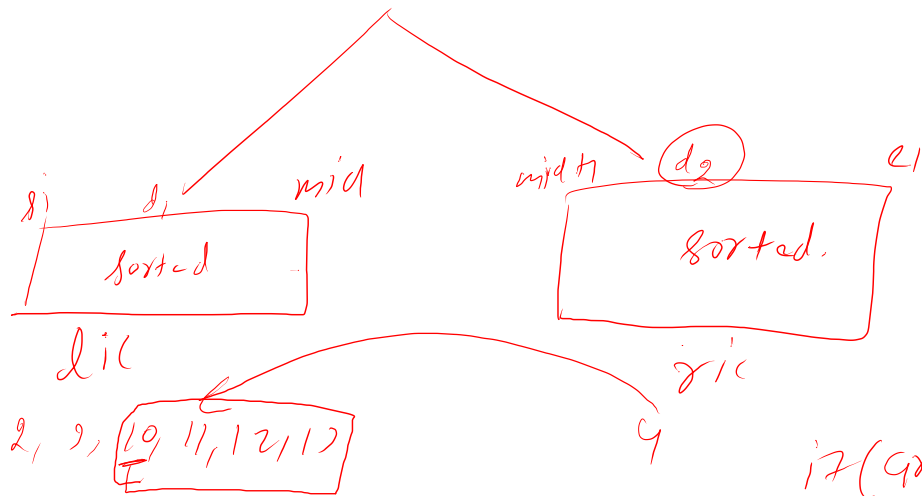
23 = 15 * 1 + 8

4 = 15 * 1

20 = 15 * 1 + 5

2

$arr[i] > arr[j]$
 $i < j$



$$ans = l1 + r1 + \text{inside inversion}$$

if ($arr[i] > arr[j]$):
 inside inversion $\neq (mid - i + 1)$

$$ans[k] = arr[i]$$

$$k++$$

$$i++$$

else: $ans[k] = arr[j]$

$$k++$$

$$j++$$