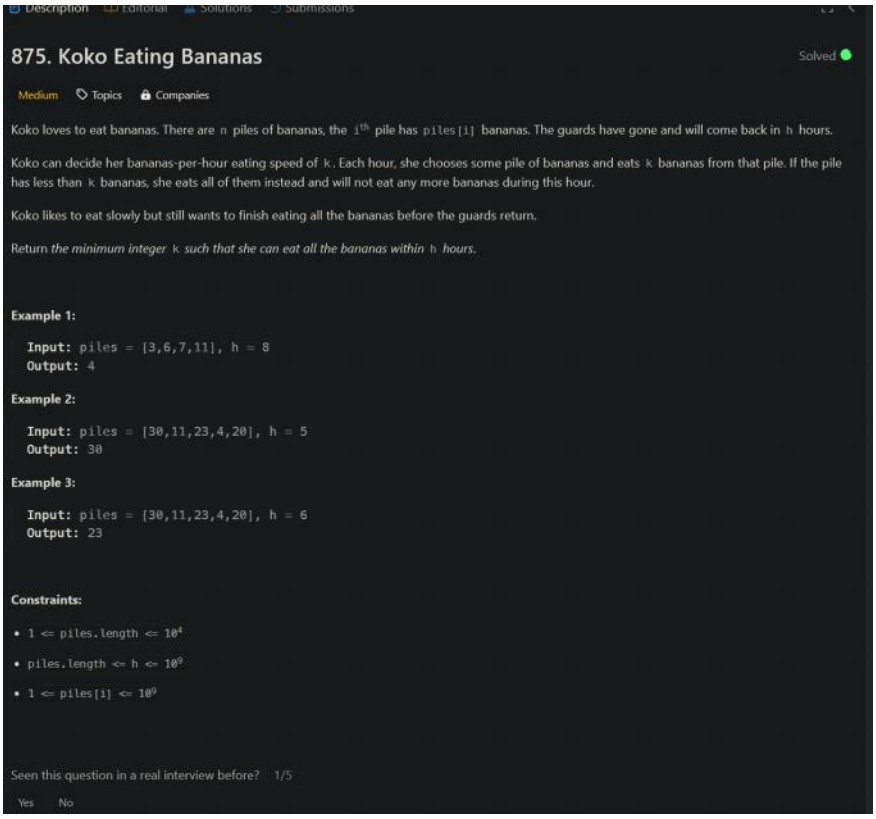
Koko eating bananas (Advanced Binary search) :-



Description Editorial Solutions Submissions

875. Koko Eating Bananas

Solved

Medium Topics Companies

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`

Seen this question in a real interview before? 1/5

Yes No

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

o/p : 4 ✓ 3 ✗

1st hour :- koko eats 3 bananas

2nd hour :- 4

3rd hour - 2

4th hour = 4

5th hour = 3

6th = 4

7th = 4

8th = 3

Let's define range for k :- 1 ⟶ max no. of bananas in pile

Let's find the test function

```
test (piles, x, h){
    int ans=0;
    for(int i: piles){
        ans += ceil( piles/x),
```

```
for(int i: piles
    ans += ceil( piles/x),
    }
  if( ans <= h) return 1;
  return 0;
  }
```

B.S.

```
int lo = 1, hi = max_piles;
while (lo < hi){
  int mid=(lo+hi)/2;
  if(test(piles, mid, h)) hi = mid;
  else lo = mid+1; }
  return lo; // required answer.
  }
```

```
C++ Auto

1  class Solution {
2  public:
3      int minEatingSpeed(vector<int>& piles, int h) {
4          int n=piles.size();
5          int hi=0;
6          for(int i:piles)hi=max(hi, i);
7          int lo=1;
8          while(lo<hi){
9              int mid=lo+(hi-lo)/2;
10             if(test(piles, mid, h)){hi=mid;}
11             else lo=mid+1;
12         }
13         return lo;
14     }
15     int test(vector<int> &piles, int k, int h){
16         int ans=0;
17         for(int i:piles){
18             ans+=((i+k-1)/k);
19         }
20         return ans<=h?1:0;
21     }
22 };
```

← T.C. $O(n \log \max(piles))$
S.C. $O(1)$

How to find ceiling of an Integer division in C++?

$ceil(a/b) = \frac{a+b-1}{b}$.