BINARY Search

DSA By Malay Tripathi

KOKO EATING BANANAS

→ Koko eating Bananas → Return the min integers K such that Koko can eat all Bananas within h hours.

K → is no. of Bananas that Koko can eat in one hr or per hours.

piles = [3 6 7 11]

h = 8 (No. of hours) Koko task is to eat Bananas at fastest pace but at max can take = 8 hours.

on first piles you have 3 Bananas
" Second " " " 6 "
" Third " " " 7 "
" fourth " " " 11 "

Koko can't move to next pile before finishing the previous one.

NOTE:- We will always take ceiling values of the hours to eat Bananas.

piles = [3 6 7 11]

K → Bananas / hr.

2 Bananas / hr.

actually take	1.5	3	3.5	5.5
ceiling value	(2)	(3)	(4)	(6)

Total hrs taken = 15 hrs.

→ So Koko end up finishing all Bananas in 15 hrs which is maximum than h = 8 hrs, which is already defined in Problem Statement.

→ Now Koko will increase value of K = 3.

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

actually time	1	2	2.33	3.66
ceiling value	①	②	③	④

10 hrs.

↳ Koko will take 10 hrs to finish all the Bananas, but h = 8. So it will take 10 hrs, not acceptable.

→ Now Koko will increase value of K = 4

piles =	3	6	7	11
actual time =	.7	1.5	1.7	2.7
ceiling	1	2	2	3

1 2 2 3 = 8 hrs.

- Now the point is can we further increase the value of K to 5, 6, 7, ... Yes but it will reduce the time taken to finish. But it will be no use. As the problem state that we have find "minimum value" of K.

So "min. integer" is **K = 4**.

BRUTE FORCE

- Starts with ONE BANANA PER HRS. AND FIND OUT TOTAL TIME, IF `TIME > h` TOTAL TIME EXCEEDS THE DEADLINE INCREASE THE COUNT. TO REDUCE TOTAL TIME TAKEN TO LESS THAN OR EQUAL TO THE DEADLINE (h).

- piles[] = [3 6 7 11]

  So "MAX. VALUE OF k" that can be taken is h = 11.

  WE CAN ALSO TAKE 12, 13, 14 .. BUT WE CONSIDER LOWEST VALUE OF THE HIGHER CONSIDERATION.

  MEANS MAX. NO. OF BANANA IN ANY PILES IS THE NO. OF BANANAS THAT YOU CAN EAT PER HRS.

- SO I THING IS SURE MY ANSWER IS ALWAYS LIES B|W [1 → 11].

can replace this LINEAR SEARCH WITH BINARY SEARCH.

```
for(i=1, → max(ARRAY))
  {
    req. TIME = fun(arr, i)
   if (reqtime <= h)
       return i;
  }
```

Gave you TIME LIMIT Exceed

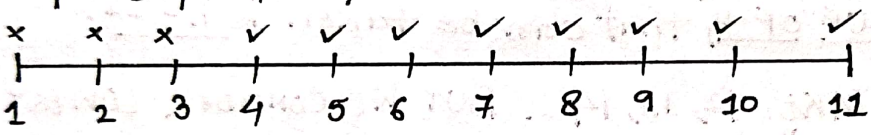
T.C = O(max(Array) x n)

↳ max. of array element multiply n.

```
fun( arr, hourly)
  {
   total hrs = 0
    for( i = 0 → n-1)
      {
        total hrs += ceil(arr[i] / hourly);
      }
    return total hrs;
  }
```

→ WHENEVER YOU FIND A PATTERN, APPLY BINARY SEARCH.

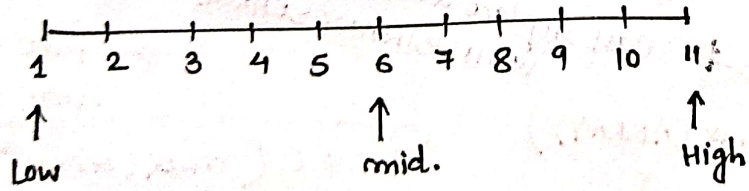
4 = LEAST VALUE AT WHICH → COMPLETION IS POSSIBLE.



Whenever you are Sure, than always apply B.S on the pattern exists.

When you apply Binary Search on Answer, it is very important to find out Range.

Because it will determine the Time Complexity or Range.



Step 1: $mid = \frac{1+11}{2} = 6$, it will take piles = [3 6 7 11] / [1 1 2 2] = 6 hrs. It is under deadline h = 8.

Yes, ans = 6. Ques. is to find min. value of k.

Pseudo Code.

```
BinarySearch (arr, h)
{
 low=1, high = max_element_in_array(arr);
 while (low <= high)
    {
     mid= high + low
          2
      total hrs = func(arr, mid);
      if (totalhrs <= hrs)
         {
           ans= mid;
           high = mid-1;
         }
      else
         {
           low = mid+1;
         }
}
```

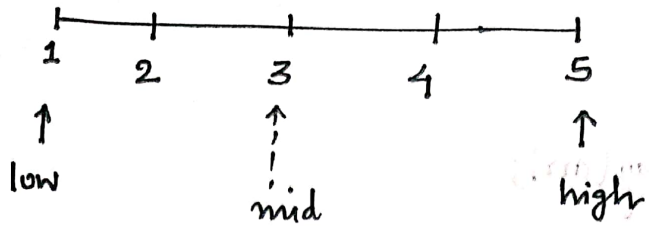
func. counting total no. of hrs.

TIME COMPLEXITY = $O(N) * \log_2(\text{max ele})$

max. element is 11, so it will take $\log_2(11)$.

DSA by Mallay Tripathi

Step 2:



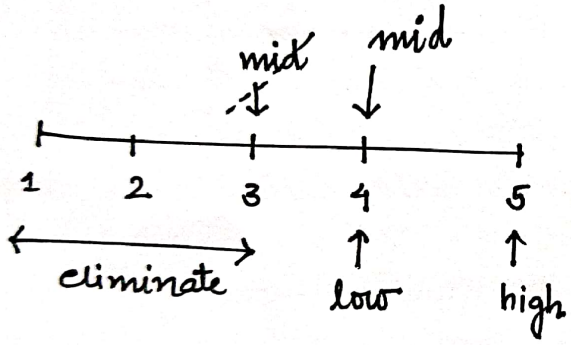
$mid = \frac{1+5}{2} = \frac{6}{2} = 3$

Piles = [3 6 7 11]

1hrs 2hrs 3hrs 4hrs = 10hrs.

Total Time > h

Step 3:



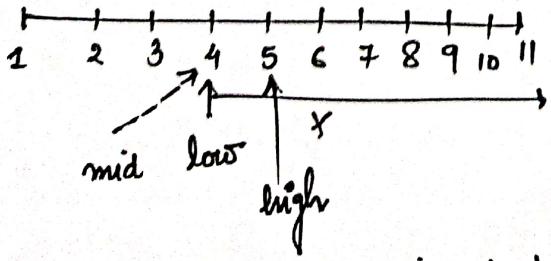
$mid = \frac{4+5}{2} = 4$

piles = [3 6 7 11]

1 2 2 3 = 8hrs, this value is equal to deadline

ans = ~~6~~ 4 (value updated to 4).

Step 4:



all values beyond 4 will also be an answer. So high move to mid-1, and then high < low, thus BINARY SEARCH STOPS. AND RETURN LOW.