875. Koko Eating Bananas

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

k -> bananas per hour eating speed.

ans = ~~8~~ 4

piles : [3, 6, 7, 11]

0 1 2 3

1 2 2 3

m = 4

4

~~3~~

lo = ~~0~~ , hi = ~~11 4~~ 3

ans -> 0 to max of array

sorted : Binary search

0, 11

5 ✓ (ans)

0, 4

2 ✗

3, 4 —3 X→ 4, 4 ✓ ——→ (4, 3)

(ans) story over

```
int lo = 0, hi = max;
int ans = 0;

while(lo <= hi) {
    int mid = lo + (hi - lo)/2;

    if(isPossible(piles,mid,h) == true) {
        ans = mid;
        hi = mid-1;
    }
    else {
        lo = mid+1;
    }
}

return ans;
```

```
public static boolean isPossible(int[]piles,int sp,int h) {
    int t = 0;

    for(int i=0; i < piles.length;i++) {
        t += Math.ceil(piles[i]*1.0 / sp);
    }

    return t <= h;
}
```

ans = ~~8~~ 7

piles : [3, 6, 7, 11]
0 1 2 3

lo = 7, hi = 6

m = 7

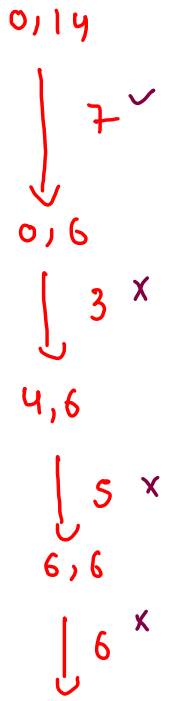
h = 5

0, 11
↓ 5 ✗
6, 11
↓ 8 ✓
6, 7
↓ 6 ✗
7, 7

piles : [3, 6, 7, 14]
0 1 2 3

h = 8

ans = 7



1283. Find the Smallest Divisor Given a Threshold

Given an array of integers `nums` and an integer `threshold`, we will choose a positive integer `divisor`, divide all the array by it, and sum the division's result. Find the **smallest** `divisor` such that the result mentioned above is less than or equal to `threshold`.

Each result of the division is rounded to the nearest integer greater than or equal to that element. (For example: `7/3 = 3` and `10/2 = 5`).

The test cases are generated so that there will be an answer.

h = 8

nums : [3, 6, 7, 11]

0 1 2 3

1 2 3 4

divisor = 4

ans: 1 to max of arr.

Allocate Minimum Number Of Pages

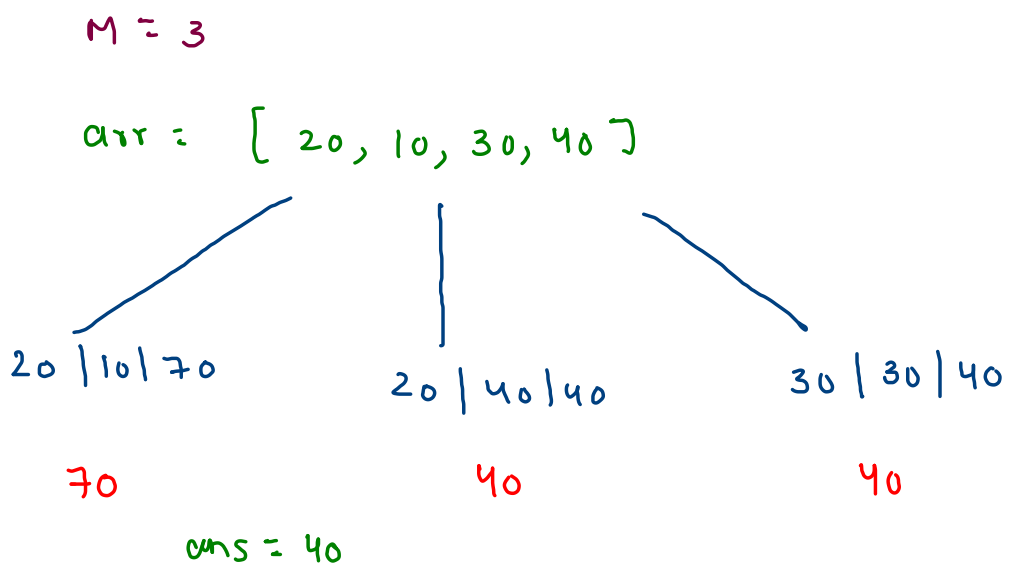
1. You are given N number of books. Every ith book has Ai number of pages.
2. You have to allocate books to M number of students. There can be many ways or permutations to do so. In each permutation, one of the M students will be allocated the maximum number of pages. Out of all these permutations, the task is to find that particular permutation in which the maximum number of pages allocated to a student is minimum of those in all the other permutations and print this minimum value.
3. Each book will be allocated to exactly one student. Each student has to be allocated at least one book.
4. Note: Return -1 if a valid assignment is not possible, and allotment should be in contiguous order.

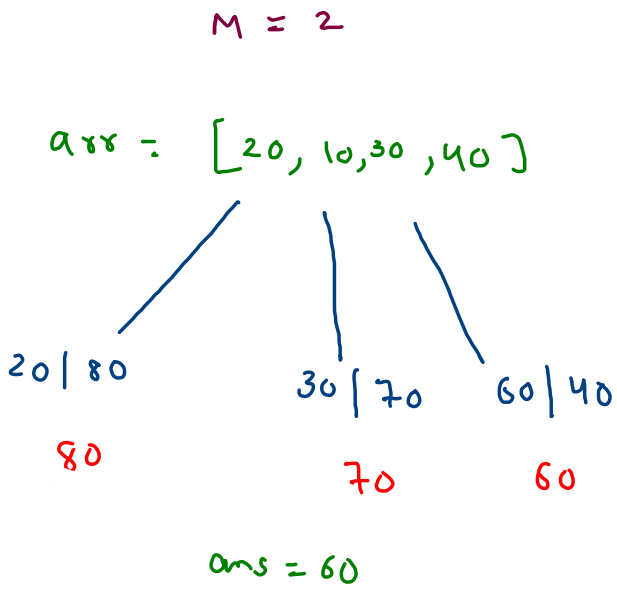
M = 2
↳ no. of students

arr : [20, 10, 30, 40]

(i) unbreakable
(ii) each student must have atleast 1 book.
(iii) allocation should be in continous manner.

2. You have to allocate books to M number of students. There can be many ways or permutations to do so. In each permutation, one of the M students will be allocated the maximum number of pages. Out of all these permutations, the task is to find that particular permutation in which the maximum number of pages allocated to a student is minimum of those in all the other permutations and print this minimum value.





arr = [20, 10, 30, 40]

M = 3

lo = 40 , hi = 39

m = 40

ans : 40 to 100

↳ sorted

↳ binary search

ans = ~~70~~ ~~54~~ ~~46~~ ~~42~~ 40