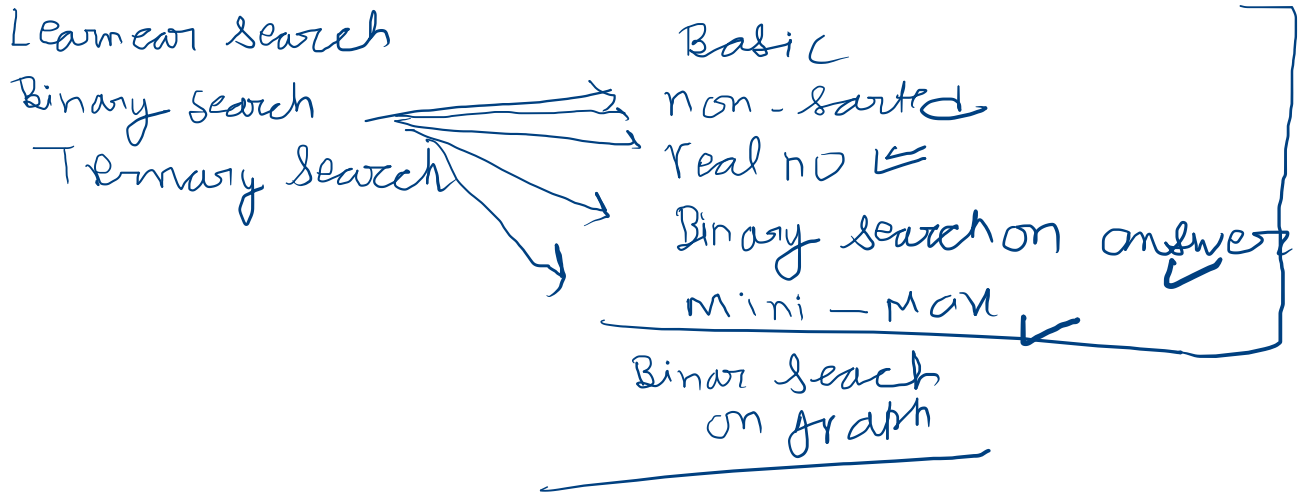
Searching


Learnear Search
Binary Search
Ternary Search
Basic
non - sorted
real no
Binary search on answer
Mini — max
Binar Seach
on graph

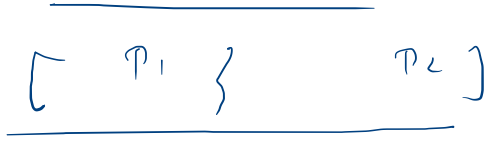
$T(n) = n - n-1 - n-2$ →

$T(n) = T(n-1) + O(1)$ → comparison

$T(n) = O(n)$

Binary Search →

Binary search says that if you have a search space at size n then we can divide the space in n/2 - by discarding 1 half



$T(n) = T\left(\frac{n}{2}\right) + O(1)$ → comparison

$T(n) = T(n/2) + O(1)$

$T(n/2) = T(n/4) + O(1)$

$T(n/4) = T(\quad) + O(1)$

$T(n) = T(1) + O(1)$

$n \to \frac{n}{2} \to n/4 \to n/8 \cdots$

$\frac{n}{2^k} = 1 \qquad n = 2^k$

$k = \log_2 n$

Sorted

Ques → find the first index that is greater or equal target?

ex → 0 1 2 3 4 5 6

10 20 30 30 70 40 50

target = 30

ans = 2

→ lower bound

→ upper bound

target = 6

(2 3 5 6 6 6 8)

0 1 2 3 4 5 6

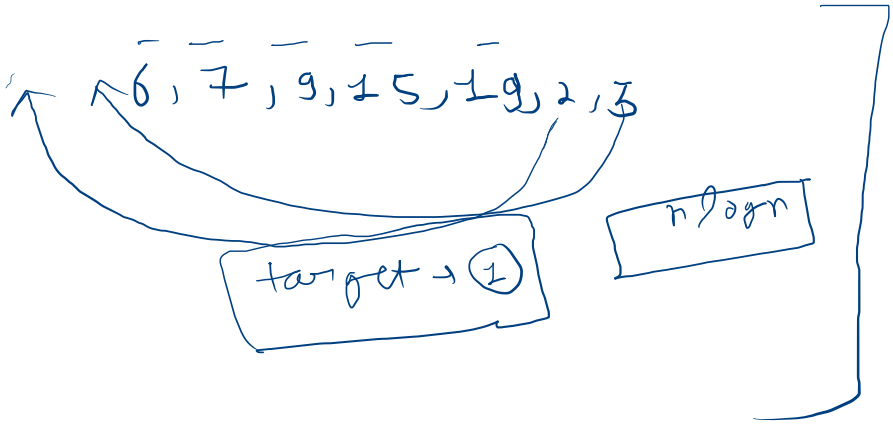
F F F T T T T

→ n/2 → $lo + \frac{(hi - lo)}{2}$

$\frac{lo + hi}{2} = \frac{lo + hi + lo - lo}{2} = \frac{2lo + hi - lo}{2} = \frac{2lo}{2} + \frac{hi - lo}{2} =$

$lo + \frac{(hi - lo)}{2}$

Que → rotated sorted array find the index of target



Que → Given an array, find any element in array such that it follows the following condition

$a[i] > a[i+1]$ (if $i+1$ exist)

and

$a[i] < a[i-1]$ (if $i-1$ exist)

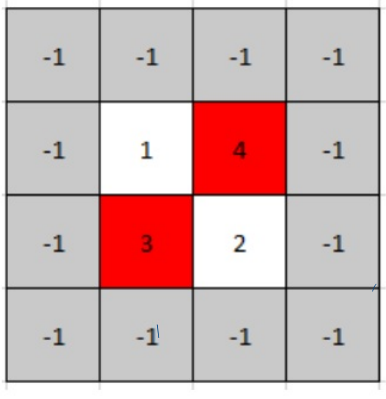
row 2 .

278. First Bad Version

Easy 5278 1963 Add to List Share

Ques Given a 2D array, find the Peak element in a 2D array (no adjacent element is equal)

Example 1:




```
Input: mat = [[1,4],[3,2]]
Output: [0,1]
Explanation: Both 3 and 4 are peak elements so [1,0]
and [0,1] are both acceptable answers.
```

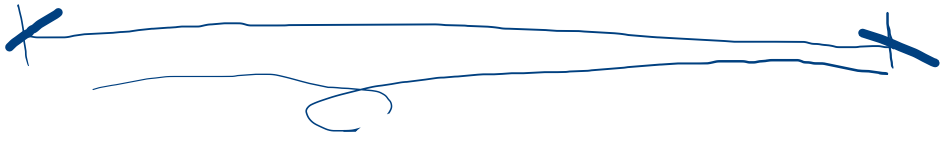
Ques → Given a number n, ($n \leq 10^5$) find the square root of n (only int value)

don't use inbuild library

$O(n^2)$ $O(n$ loops$)$

Binary search on answer

$n = 36 = 6$

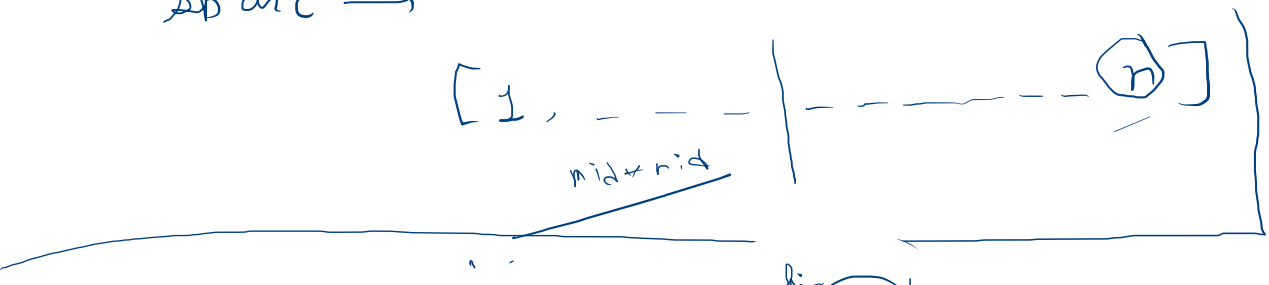


→ if the value of mid * mid, can i say the sqrt is ≤ n

→ value of n is Positive

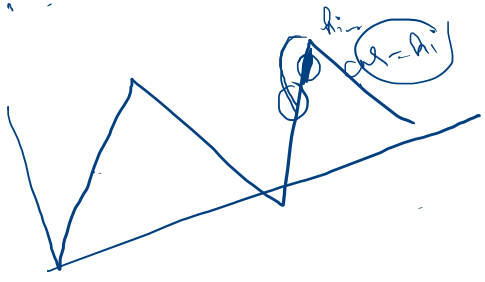
14

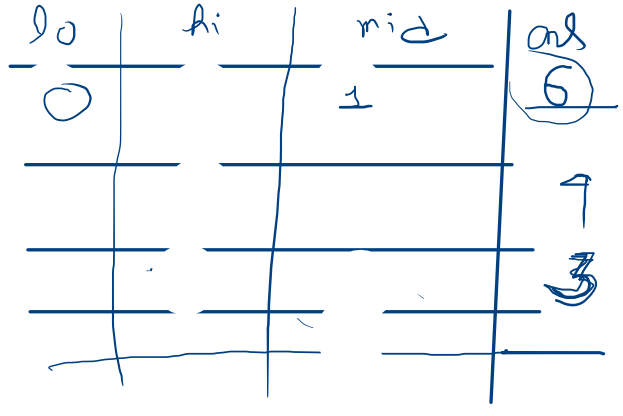
Space →




```
int lo=0;
int hi=n;
int ans=0;
while(lo<=hi){
    int mid=lo+(hi-lo)/2;
    if(mid*mid==n){
        return mid;
    }
    else if(mid*mid<n){
        //discard the left half
        lo=mid+1;
    }
    else{
        hi=mid-1;
        ans=hi;
    }
}
return ans;
```

3/2





There are n rectangle of same size (w * h)

find a Square of smallest size into which all of the n rectangle

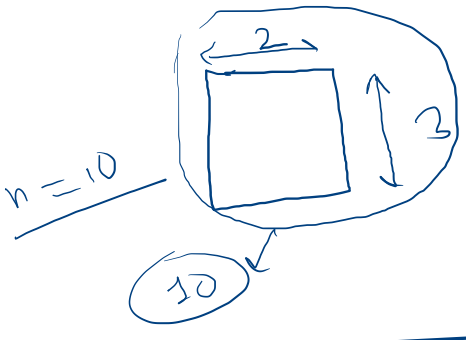
can be placed (Rotation is not allowed)

$(w, h, n) \leq 10^9$

$w = 2 \quad h = 3 \quad n = 10$

ans = 9

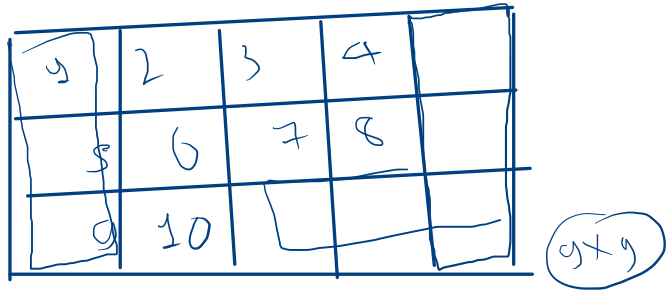
w := h

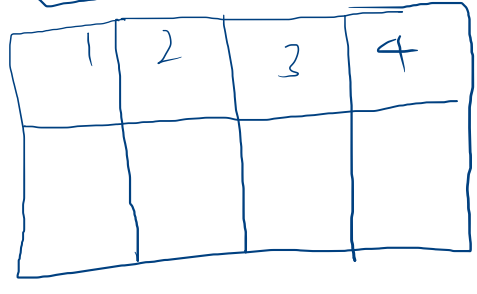


w, h → 3 * 2 = 6 area

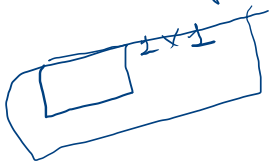
10

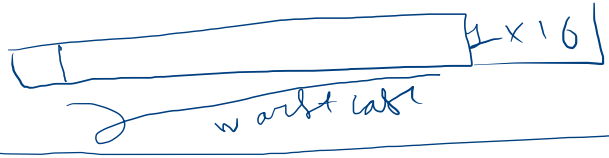
overall = 60

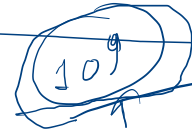


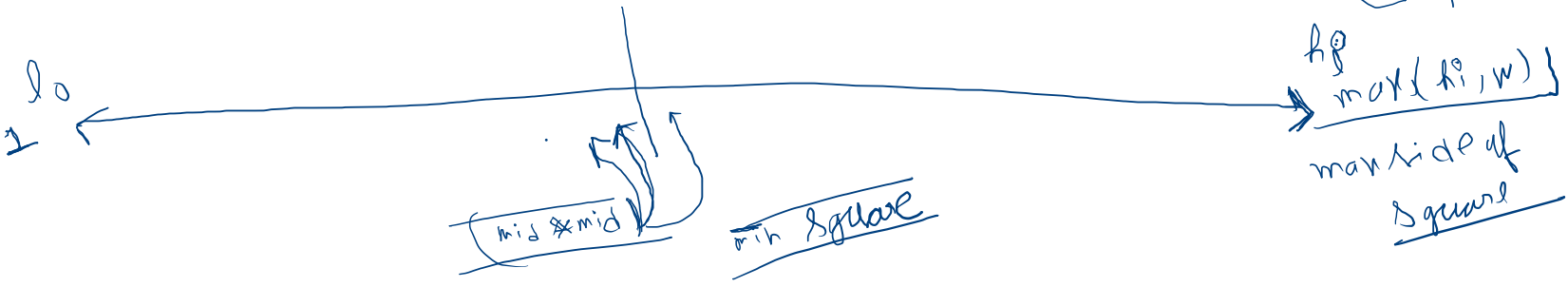


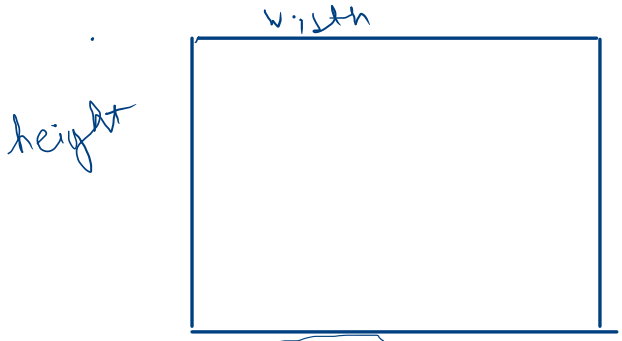
One of the sides of rectangle will cover a square side completly



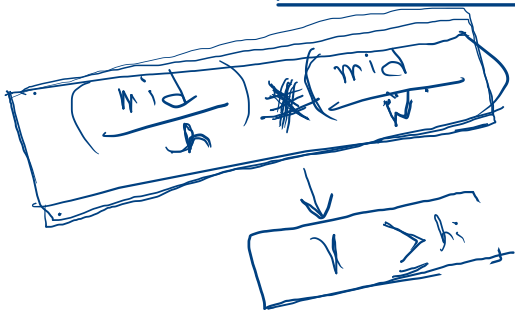


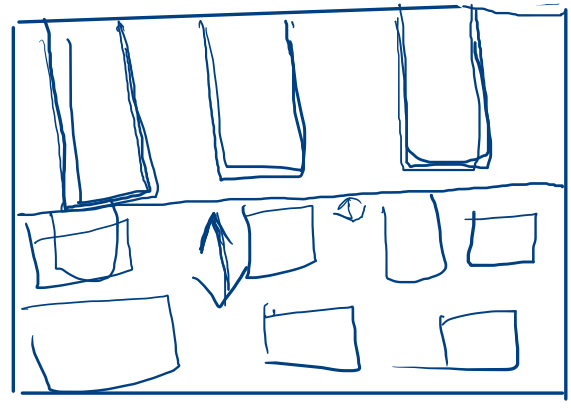






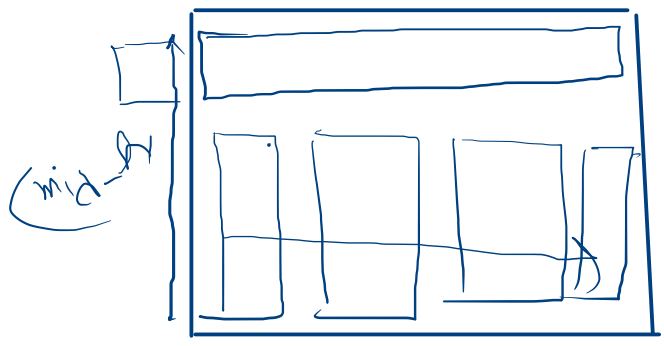
(mid * mid) square can fit horizontally ① vertically





(mid / h)

horizontally → (mid / w)



① horizontally

(mid) / h

② vertically

(mid) / w

(mid) * (mid / w) >= h

we can fit

n rectangles

Q → You want to a Photocopy shop which has got 2 machine. first takes x sec to copy a sheet of paper & another takes y sec to copy a Paper. Both can be used parallel, you have a Peice of Paper for which you need n copies. find the min time required to to print all Pages

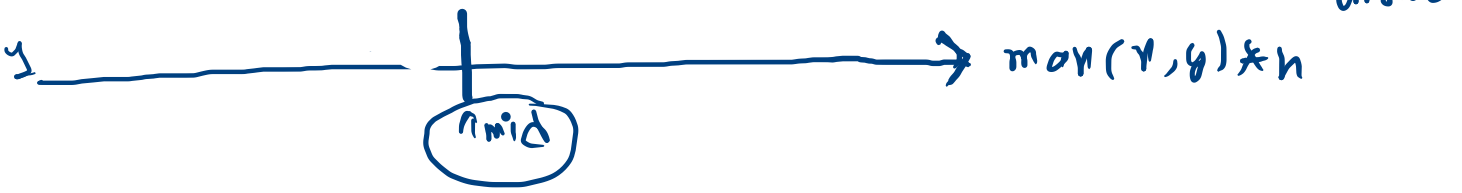
n = 5, x = 1, y = 2 → ans = 4

$n \leq 10^6$

int = total = 5

max(x,y) * n

→ Binary search on answer



n copies

1 copy → $\min(x, y)$

to run both machine Parallely we need to make atleast one copy

A → 1 copy → x sec

1 sec = $\frac{1}{x}$ copy

mid sec = $\frac{mid}{x}$ copy

B → 1 copy → y sec

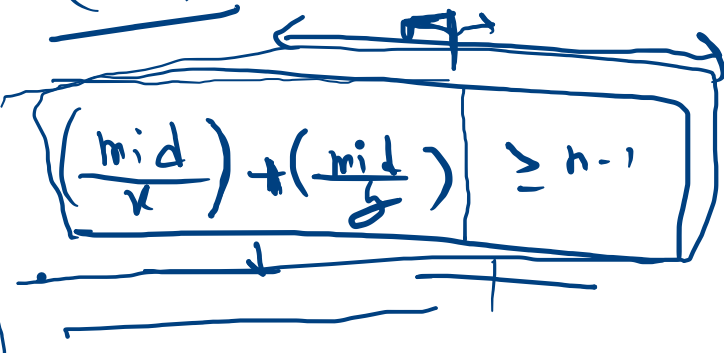
1 sec = $\frac{1}{y}$ copy

mid sec = $\frac{mid}{y}$ copy

$\left(\frac{mid}{x}\right) + \left(\frac{mid}{y}\right) \geq n-1$

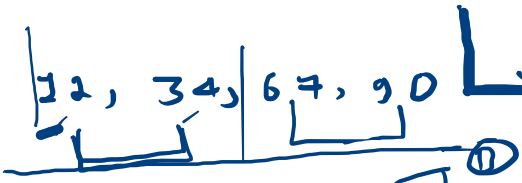
1 → $\min(x, y)$

$(n-1)$



Binary search on mini– max

Ques You are given no of pages in a Book and we have m student Books are arranged in sorted order based on no of pages Devide the book among m student such that maximum Page a student has to read is minimized



A

max ([12], [34, 67, 90]) = 191 ↓

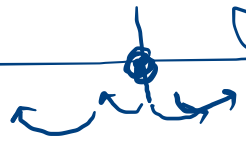
max(12, 191) = 191 ↓

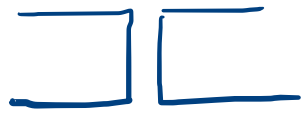
max ([12, 34), (64, 90)) = 154 ↓

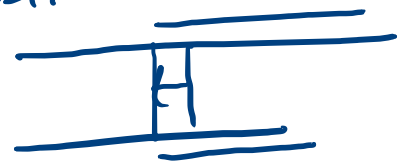
(12, 34, 67) (90) =

(113, 90) = 113 ↓ min

max(arr[i]) ——— $\sum_{i=0}^{n}$ (arr[i])







mid, m

[11, 34, 67, 90]

```
Possible(arr, mid, m) {
  int s = 1;
  cur_sum = 0;
  for(int i = 0; i<n; i++) {
    if(cur_sum + arr[i] > mid) {
      s += 1;
      cur_sum = arr[i];
    }
    else
      cur_sum += arr[i];
  }
  if(s > m) return false;
  return true;
}
```

Ans: Aggressive Conv (Proj)