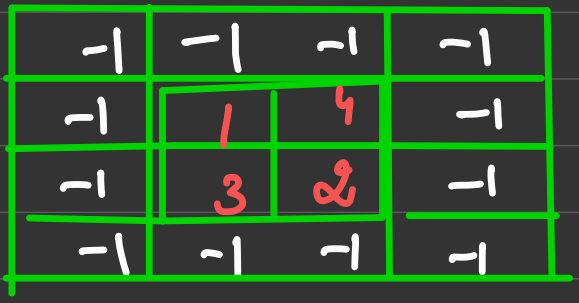
Q:) Given a 2D array, find the peak element in this 2D array. (no adjacent element is equal)



n x m

3 or 4

return any one peak element.

$n, m \leq 10^4$

Brute force → to find the max element of the grid. → O(nm)

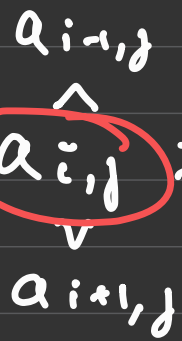
Can we optimize ??

Column max elements also

4	2	5	1	4	5	→ 5
2	9	3	2	3	2	→ 9
1	7	6	0	1	3	→ 7
3	6	8	3	7	2	→ 7

m - rows
n - colums

Peak element → $a_{i,j-1} < a_{i,j} > a_{i,j+1}$



$O(n,m)$

$O(n \log m)$

Can we extract some info column-wise or rowwise, that can simplify the problem:

How abt we calc maximum of any one row??

i → row under inspection j → idx of max element

```
if( arr[i][j] < arr[i+1][j] ) inc
    10<=i<=1
else de
    → hi=i
```

Q: Given a number n, ($n \leq 10^{15}$) find the square root of n (only integer value)
Don't use inbuilt library.

$n = 36 \longrightarrow$ ans $\rightarrow 6$

Binary Search On ans

→ if the value of no. is n, can I say, the sqrt <= n.

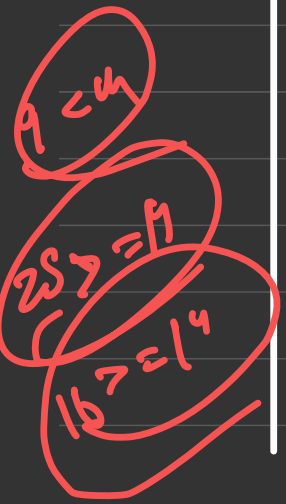
→ whatever is the value of n (positive) sqrt >= 1

→ sqrt lies in the range [1, n]

Search problem → x (largest x)

$x * x \leq n$

	lo	hi	mid	ans
1st	0	14	7	14
	0	6	3	0
	4	6	5	4
	4	4	4	3



Q:- There are n rectangles of same size (w × h) w → width h → height

find a square of smallest size into which all of the n rectangles can be packed. (Rotation is not allowed)

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

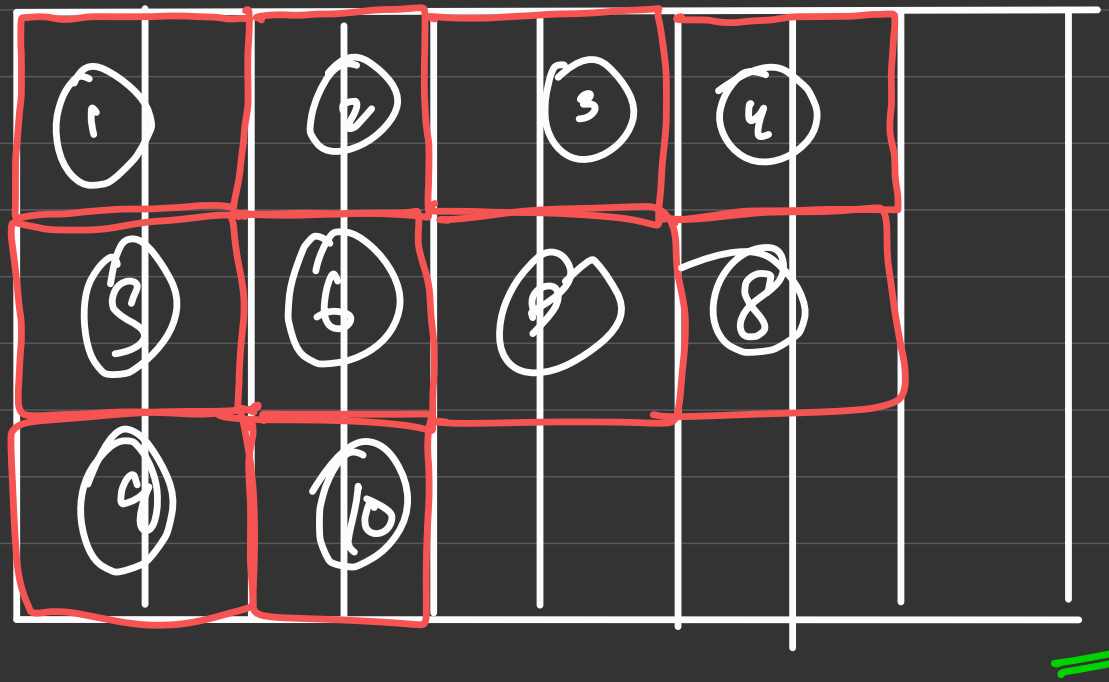
w = 2 h = 3 n = 10

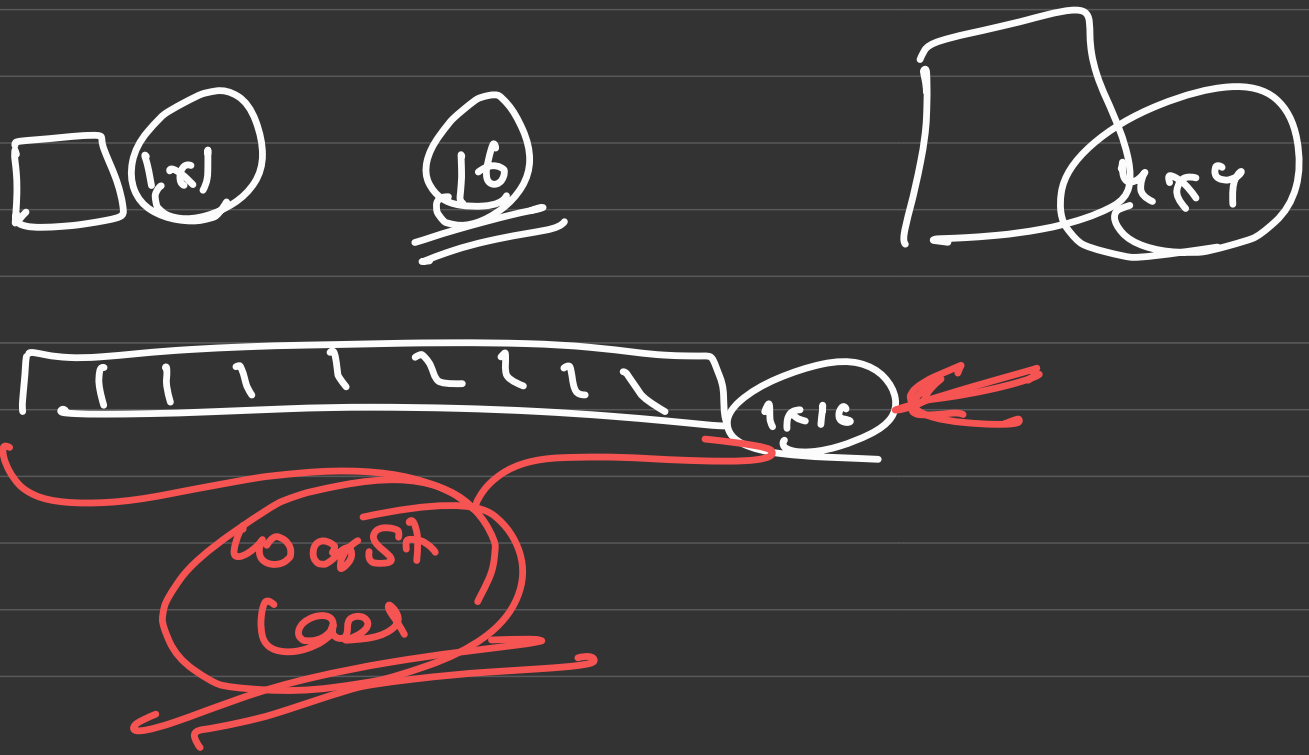
ans → 9

rectangle

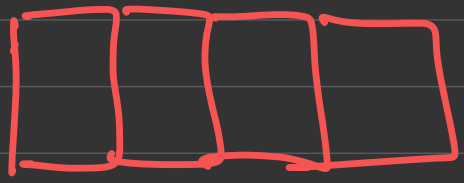
Pièce

n=10



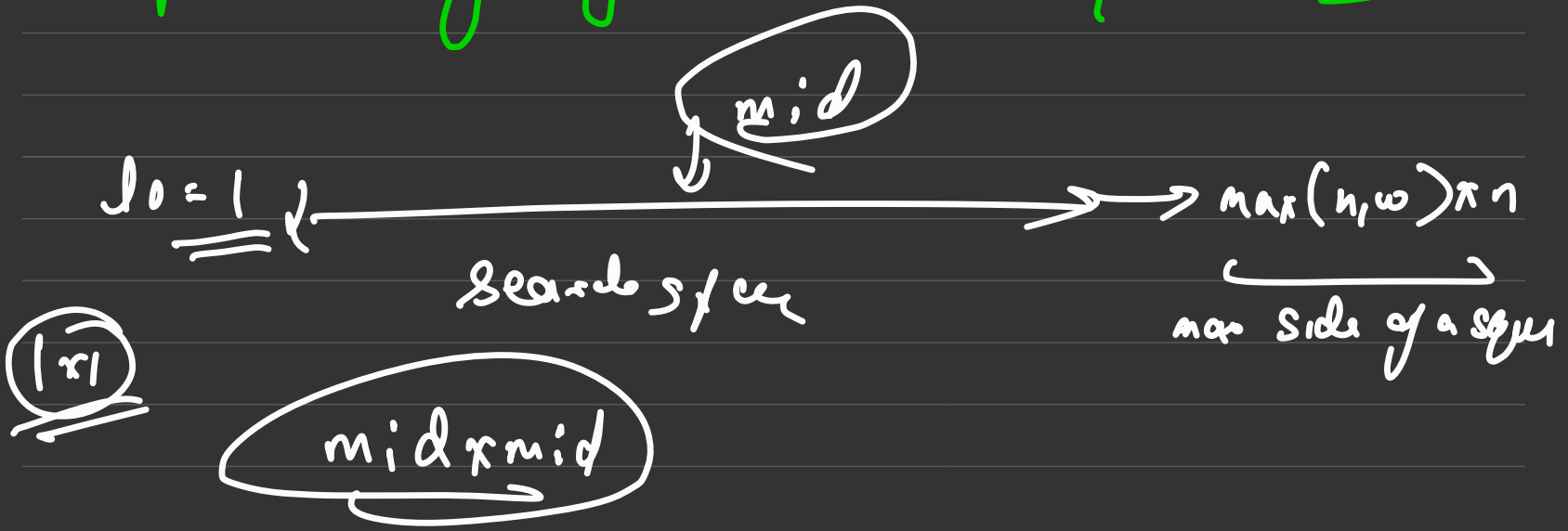


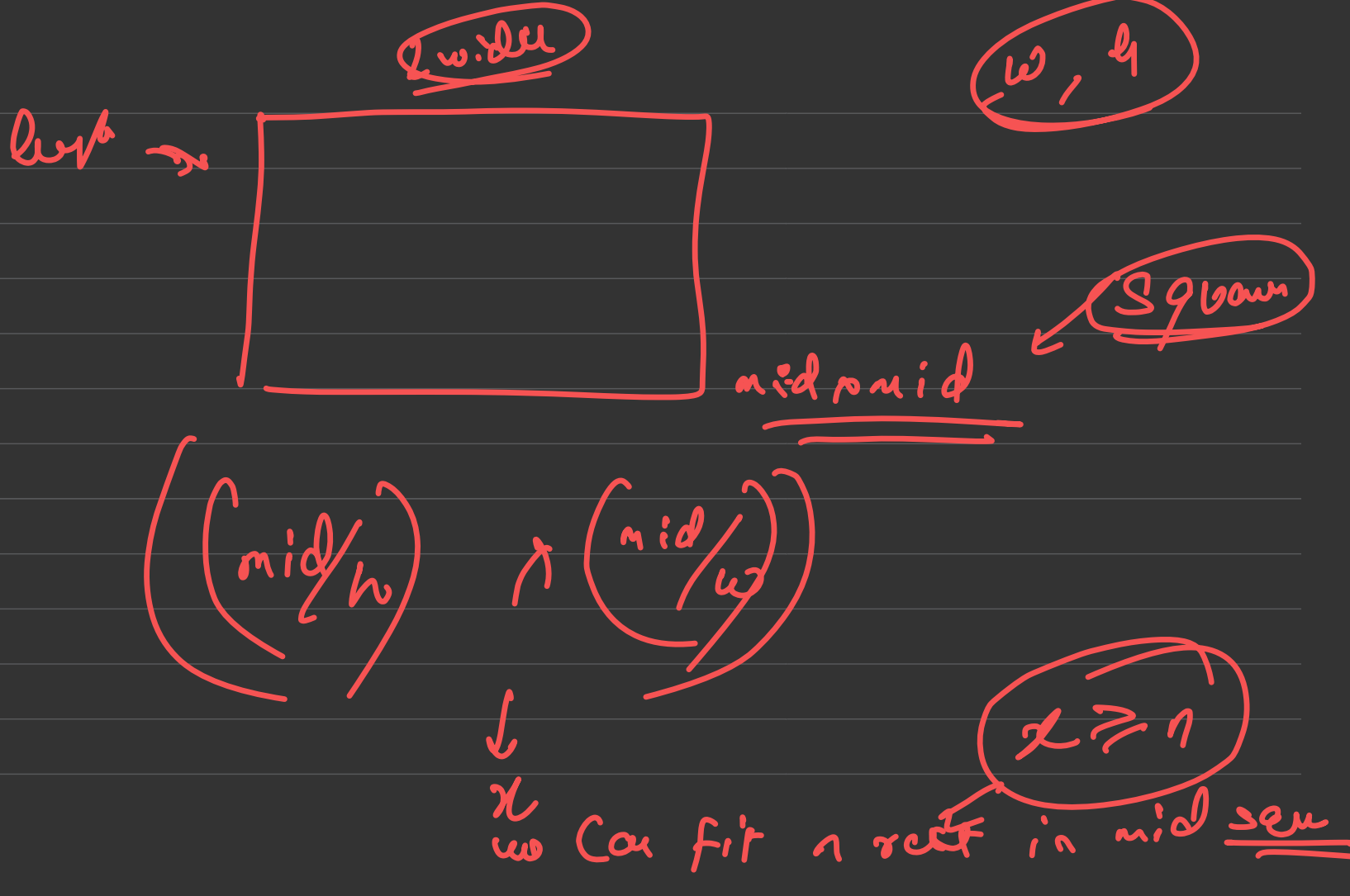
w != h



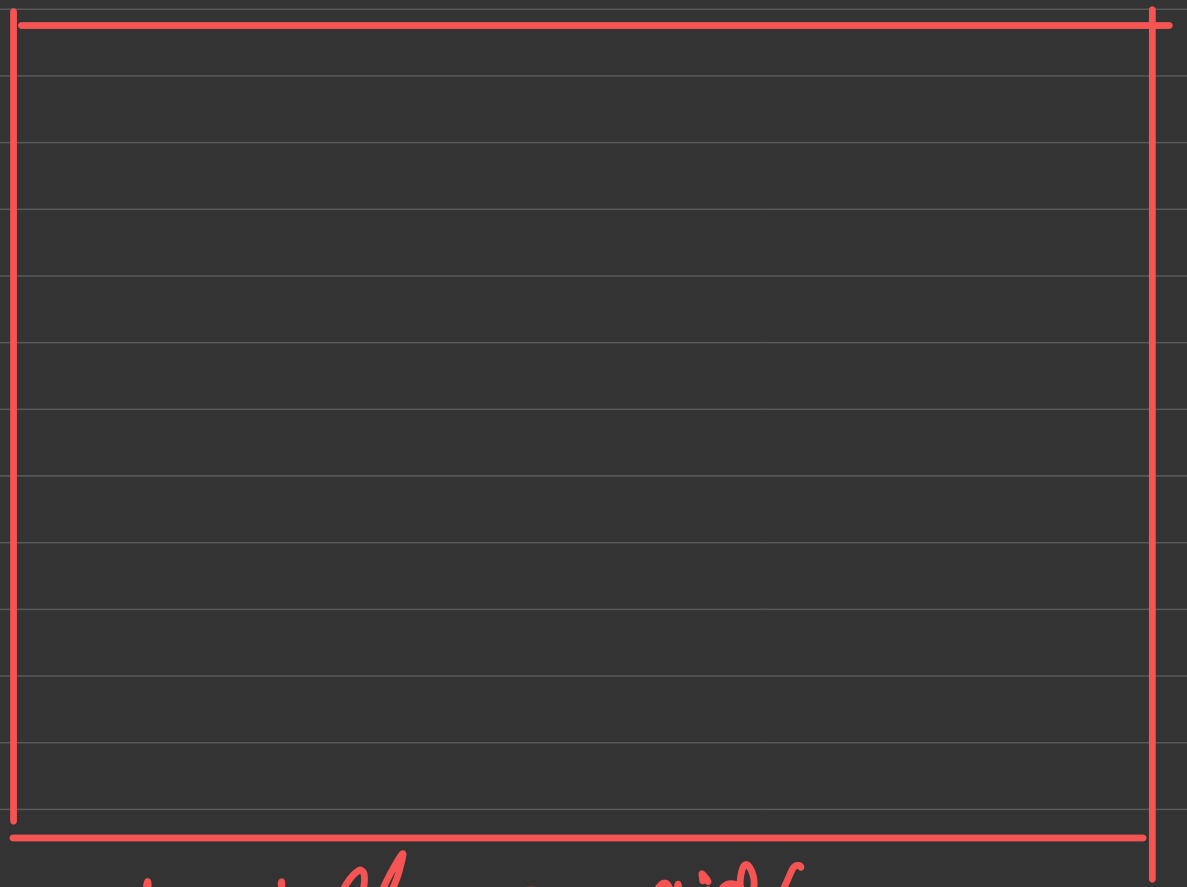
One of the sides of rectangle will cover a screen side completely.

In the worst case we can conclude that our bigger side of rectangle will be confined highly under a square side




w, h
Square
mid * mid
(mid/h) * (mid/w)
x
w Can fit * rect in mid sqr
x >= n

How many rect we can fit ① → vertically → $\frac{mid}{h}$



② horizontally → mid/w

Q. You went to a photocopy shop which has got 2 machines. first machine takes x sec to copy one sheet & other takes y seconds to copy one sheet. Both can be used parallely. You have a piece of paper for which you need n copies. find the min time reqd.

n = 5 x = 1 y = 2

ans → (1)

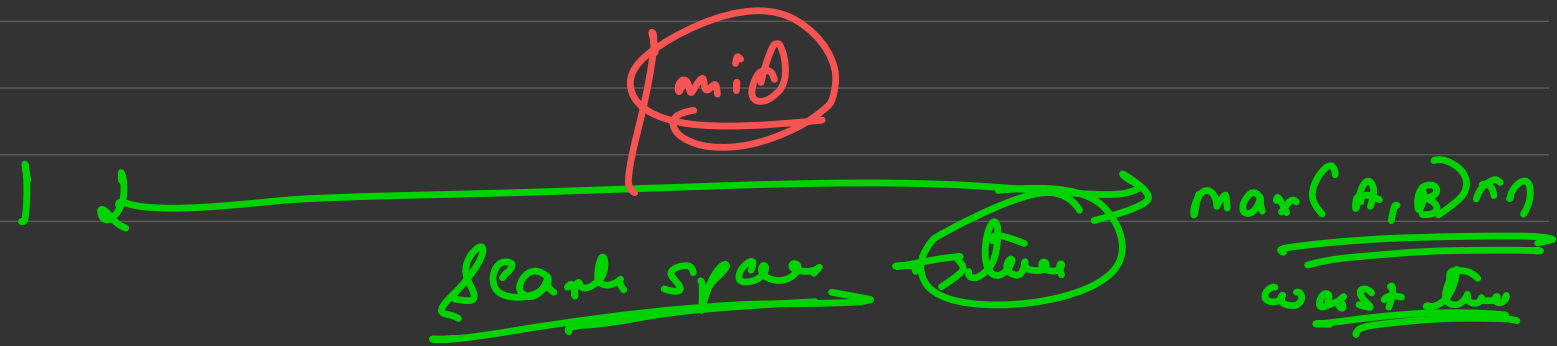
→ n more copies

A B

1 orig down

1 copy → min(A, B)

To run both parallely, we need to make atleast one copy:



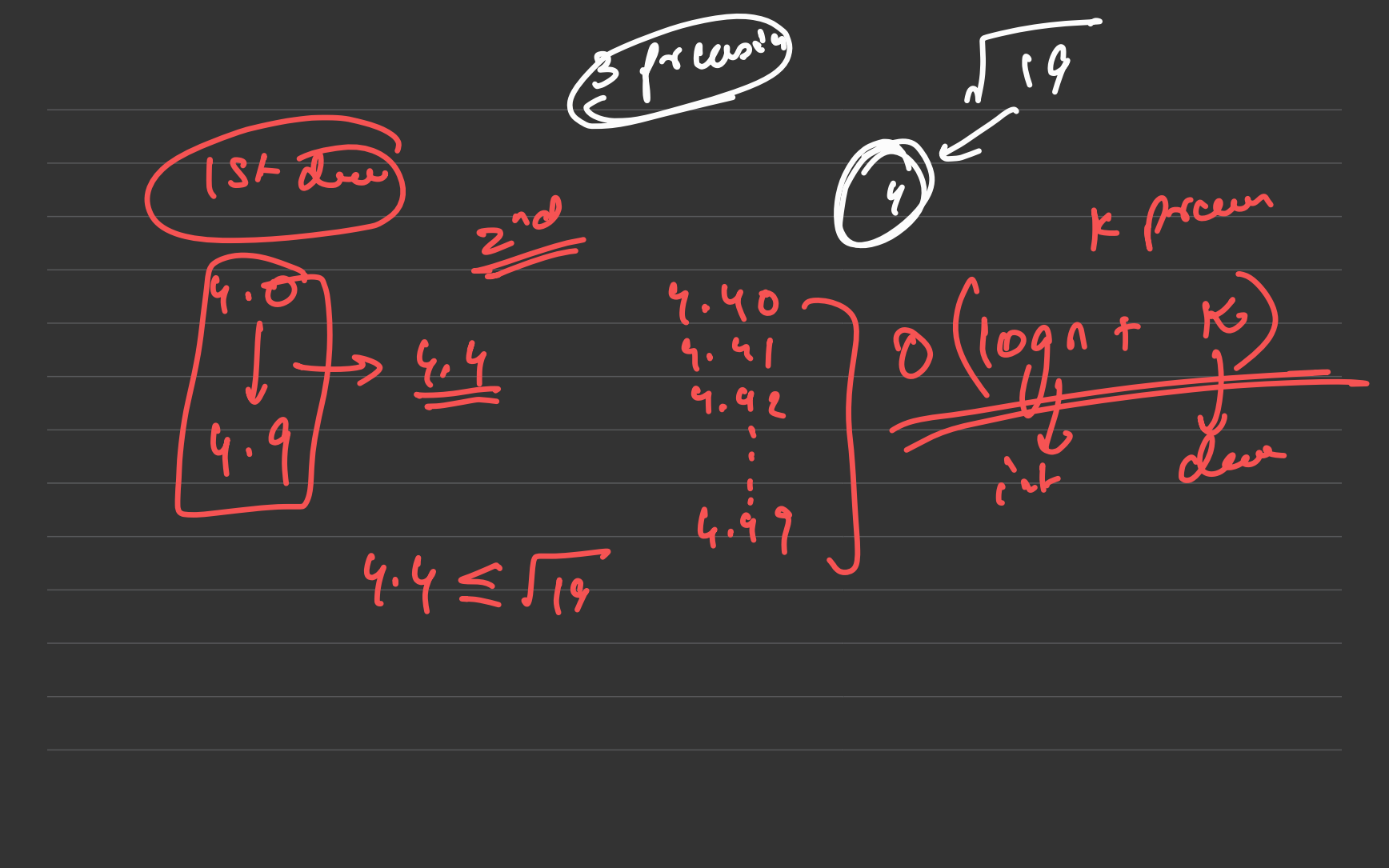
A → 1 copy → x sec

1 sec → $\frac{1}{x}$ copy

mid sec → $\frac{mid}{x}$ copy

B → mid sec → $\frac{mid}{y}$ copy

$\frac{mid}{x} + \frac{mid}{y} \geq n-1$


3 precision
√19
4
1st digit
2nd
K precision
4.0
4.9
4.4
4.40
4.41
4.42
4.99
O(log n + K)
int
digit
4.4 ≤ √19