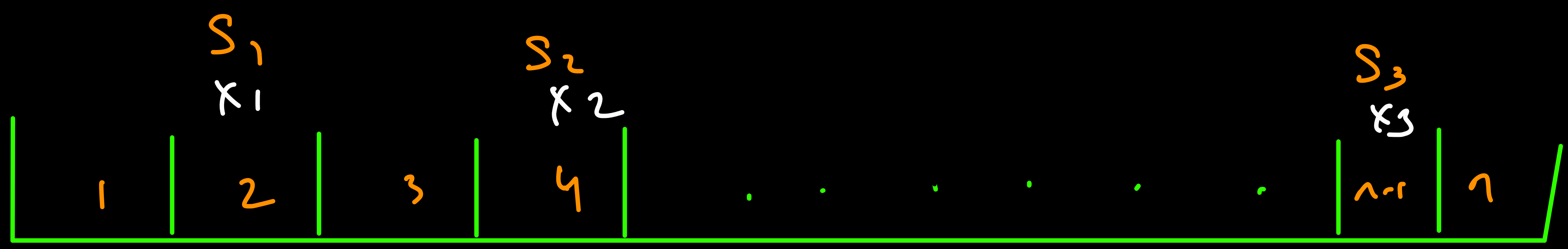
Minimax Problems Of Searching

Minimize the maximum of a value

OR

Maximize the minimum of a value.

$N \leq 10^5$



C - cows

minimum dist btw any 2 cows is maximized ans = ~~8~~ 3

C(1, 2, 8, 4, 9)

C = 3

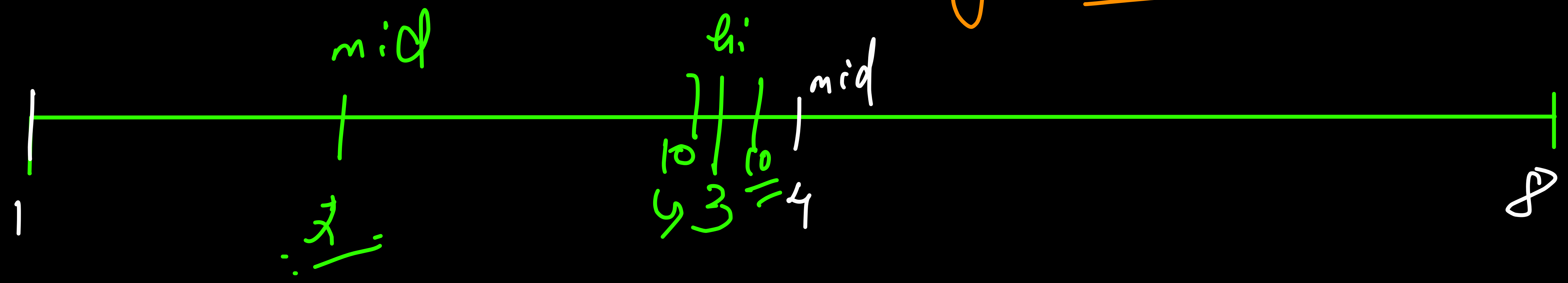
C3

C3

C1		C2	C3	
1	2	4	8	9

→ candi → 1, 3, 1 . . . → 3

maximize the min dist between any 2 cows



mid → search space

min ↓ 1 <———————————> max dist

$stall_{n-1} - stall_0$

Let's try to place the cows in the barn such that any two adjacent cows have atleast mid unit of dist

↳ if we can place then mid will be a possible ans → try to find bigger than mid & move right

else
move left

$\rightarrow O\left(n\log n + n\log(max-min)\right)$

```
canPlaceCows(stalls, n, c, mid) {
    count = 1; // we can place 1 cow definitely
    last_pos = stalls[0]; // can place 1st cow on 1st stall
    for (i = 1; i < n; i++) {
        if (stalls[i] - last_pos >= mid) {
            count++;
            last_pos = stalls[i];
        }
        if (count >= c ) return true;
    }
    return false
}
```

```
lo = 1,   hi = max - min
while (lo <= hi)
    mid = lo + (hi - lo)/2
    if ( canPlaceCows ( stalls, n, c, mid) ) {
        ans = mid;
        lo = mid + 1;
    } else {
        hi = mid - 1
    }
}
return ans;
```

a_1

a_2

$O(m+n)$

merge 2 sorted arrays

7, 12, 14, 15
1, 2, 3, 4, 9, 11

1, 2, 3, 4, 7, 9, 11, 12, 14, 15

10

7, 12, 14, 15

1, 2, 3, 4, 9, 11

$4 \le 7$

0 1 2 3 4 5 6 7 8 9

$\rightarrow (a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_5 \quad a_6 \quad a_7)\ m$

$\rightarrow (b_1 \quad b_2 \quad b_3 \quad b_4 \quad b_5 \quad b_6)\ n$

→ after merging the median will be at $\left(\frac{m+n+1}{2}\right)^{th}$ index

$A \to (a_1 \quad a_2 \quad a_3 \mid a_4 \quad a_5 \quad a_6 \quad a_7)$ m

$B \to (b_1 \quad b_2 \quad b_3 \mid b_4 \quad b_5 \quad b_6)$ n

X

leftground

$a_1 \quad a_2 \quad a_3 \mid$

$a_3 \le a_4$

$b_3 \le b_4$

left

and

$a_3 \le b_4$

$b_3 \le a_4$

right

$a_3 \not\le b_4$

$a_3 > b_4$

$b_3 \not\le a_4$

$b_3 > a_4$

median

0 1 2 3 4 5 6 7 8 9 10 11 12

$N \le 100$

$-3\times10^4 \le a_i \le 3\times10^4$

$[2, 3] \longrightarrow (a, b, c, d, e, f)$

$d \neq 0$

$$\frac{a \times b + c}{d} - e = f$$

$$\frac{2 \times 3 + 2}{2} - 2 = 2 \longrightarrow (2, 3, 2, 2, 2, 2)$$

$$\frac{3 \times 2 + 2}{2} - 2 = 2 \longrightarrow (3, 2, 2, 2, 2, 2)$$

$$\frac{3 \times 3 + 3}{3} - 2 = 2 \longrightarrow (3, 3, 3, 3, 2, 2)$$

$$\frac{3 \times 3 + 3}{2} - 3 = 3 \qquad (3, 3, 3, 2, 3, 3)$$

$N \le 10^2$ Brute force → to generate all possible sextuples

$(10^2)^6 \rightarrow 10^{12}$ TLE $O(N^6)$

```
for(a=0;a<N;a++)
  for(b=0;b<N;b++)
    for(c=0;c<n;c++)
      for(d=0;d<n;d++)  → if(d≠0)
        for(e=0;e<n;e++)
          for(f=0;f<n;f++)
```

$$\frac{a \times b + c}{d} - e = f \qquad (a, b, c, d, e, f)$$

By simplification

$$\frac{a \times b + c}{d} = f + e \qquad d \neq 0$$

$$\boxed{a \times b + c = d \times (f + e)} \qquad d \neq 0$$

if we think in terms of triplets

$$((abc)\ (def))$$

[2, 3]

[(2, 2, 2) (2, 2, 3) (2, 3, 2) (2, 3, 3)

(3, 2, 2) (3, 2, 3) (3 3 2) (3 3 3)

. . . .]

LHS = [ ]   arr → [2, 3]   all possibilities of a * b + c

RHS = [ ]

```
for ( a = 0; a < n; a++ ){
    for (b = 0; b < n; b++) {
        for ( c = 0; c < n; c++) {
            LHS.push (arr[a] * arr[b] + arr[c])
        }
    }
}
```

→ $O(n^3)$

```
for( d=0; d<n ; d++) {
    for(e=0; e<n ; e++) {
        for( f=0 ; f<n ; f++) {
            if( arr[d] != 0)
                RHS.push (arr[d] x (arr[e]+arr[f]))
        }
    }
}
```

$\rightarrow O(n^3)$

[2, 3)

2 2 2   2 2 3   2 3 2   2 3 ? . . . .

a×b+c	6	7	8	9	. . . → LHS
d×(e+f)	8	10	10	12	. . . → RHS

LHS = [ ]

RHS = [ ]

for each value of LHS how many same values we find in RHS

[2:3]

[6, 7, 8, 9, 8, 9, 11, 12] → LHS $a \times b + c$

[8, 10, 10, 12, 12, 15, 15, 18] → RHS $d(e+f)$

sort ↗

[

$a \times b + c = d \times (e+f)$

ans = ~~1~~ ~~2~~ 4

lower_bound

upper_bound

$(N^3 + N^3 + N^3 \log N^3 + 2 N^3 \log N^3)$

$3 N^3 \log N^3 \approx 3 \times 3 \, N^3 \log N \rightarrow O(N^3 \log N)$

space → $O(N^3)$

n → m1 m2 x y

C. Very Easy Task

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

original → copy

c1 c2

This morning the jury decided to add one more, Very Easy Problem to the problemset of the olympiad. The executive secretary of the Organizing Committee has printed its statement in one copy, and now they need to make n more copies before the start of the olympiad. They have two copiers at his disposal, one of which copies a sheet in x seconds, and the other in y seconds. (It is allowed to use one copier or both at the same time. You can copy not only from the original, but also from the copy.) Help them find out what is the minimum time they need to make n copies of the statement.

Input

The program receives three integers n, x, and y ($1 \le n \le 2 \cdot 10^8$, $1 \le x, y \le 10$).

Output

Print one integer, the minimum time in seconds required to get n copies.

$10^8 \times 10$

$\log(10^9) \to 30$

Examples

input

```
4 1 1
```

output

```
3
```

input

```
5 1 2
```

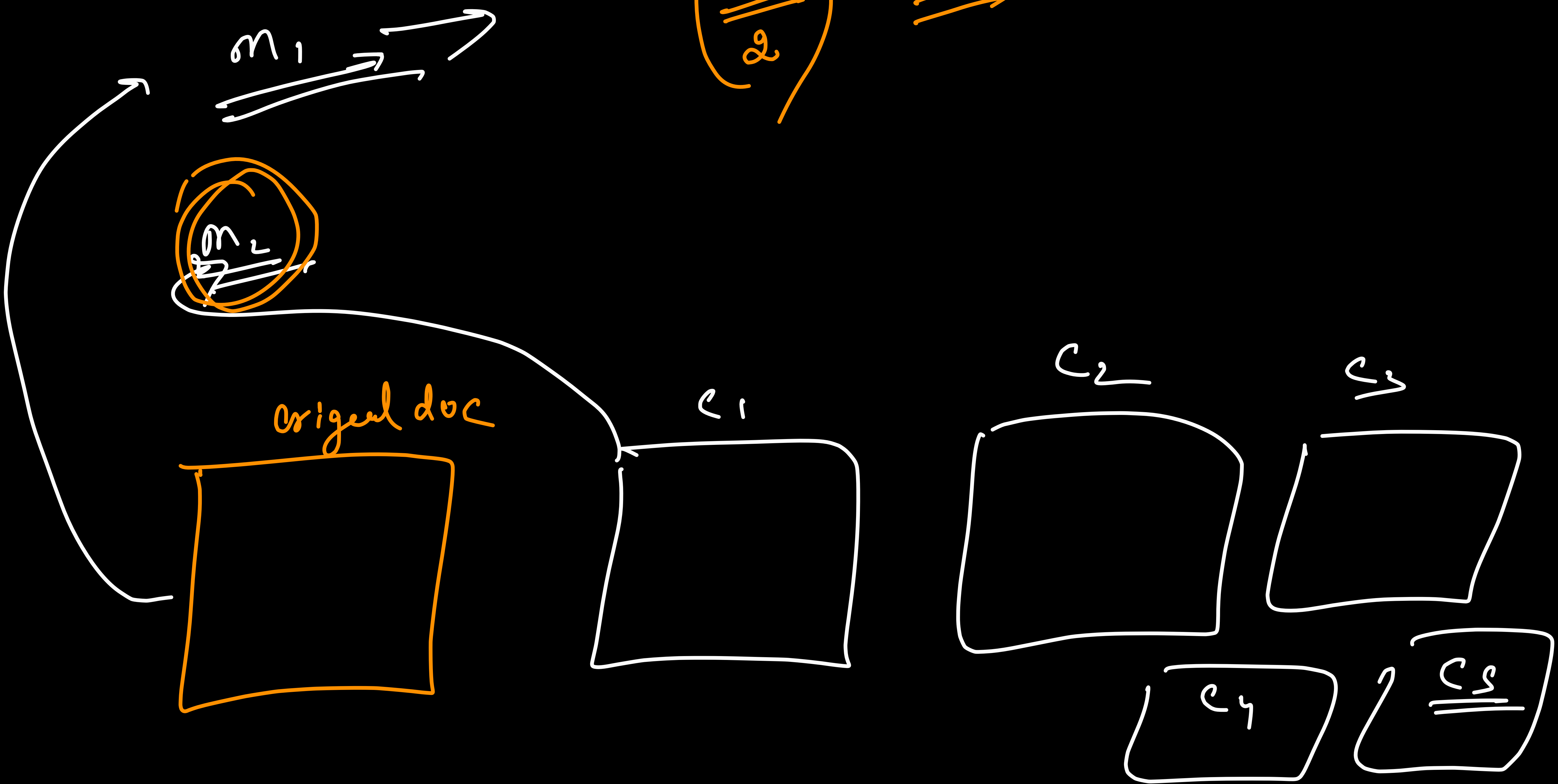
output

```
4
```

$n = 5$ $\quad x = 1$ $\quad y = 2$

$\left(\frac{10}{2}\right) \rightarrow 5$

fine → 1 2 3 4



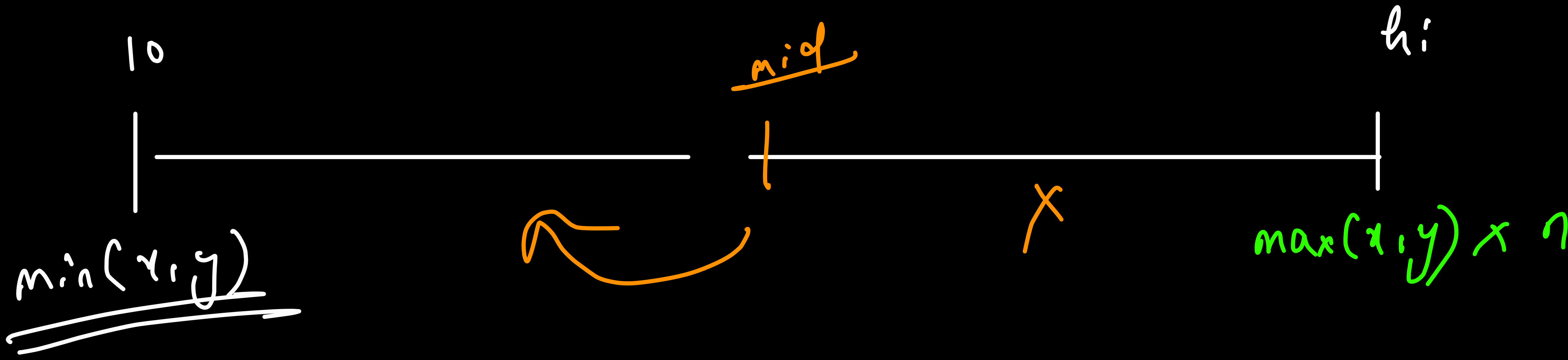
n 1^{st} copy $\rightarrow$ $\min(x, y)$

$n-1$ more copies

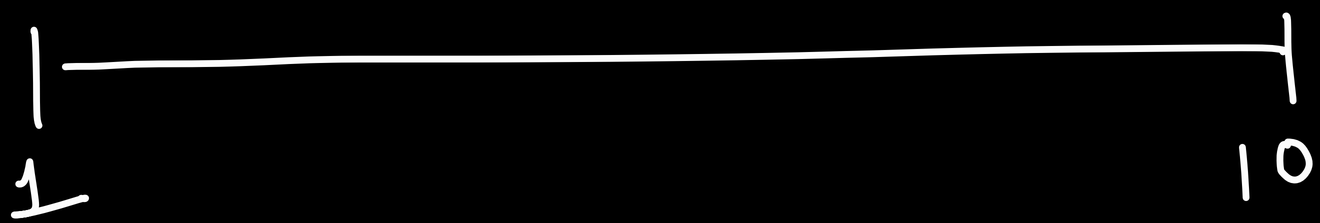
in 2 unit of time

$M_1 \rightarrow \frac{2}{x}$ copies

$M_2 \rightarrow \frac{2}{y}$ copies



n=5
x=1
y=2



n=1

Can we print n-1 papers in mid unit of time

$n_1 \rightarrow \frac{mid}{x}$

$n_2 \rightarrow \frac{mid}{y}$

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

```
ans = -1
lo = min(x, y)
hi = n x max(x, y)
while (lo <= hi) {
    mid = lo + (hi-lo)/2
    copies = mid/x + mid/y
    if (copies >= n-1) {
        ans = mid
        hi = mid-1
    } else {
        lo = mid+1
    }
}
return ans + min(x, y)
```

$O(\log(\max(x, y) \times n))$