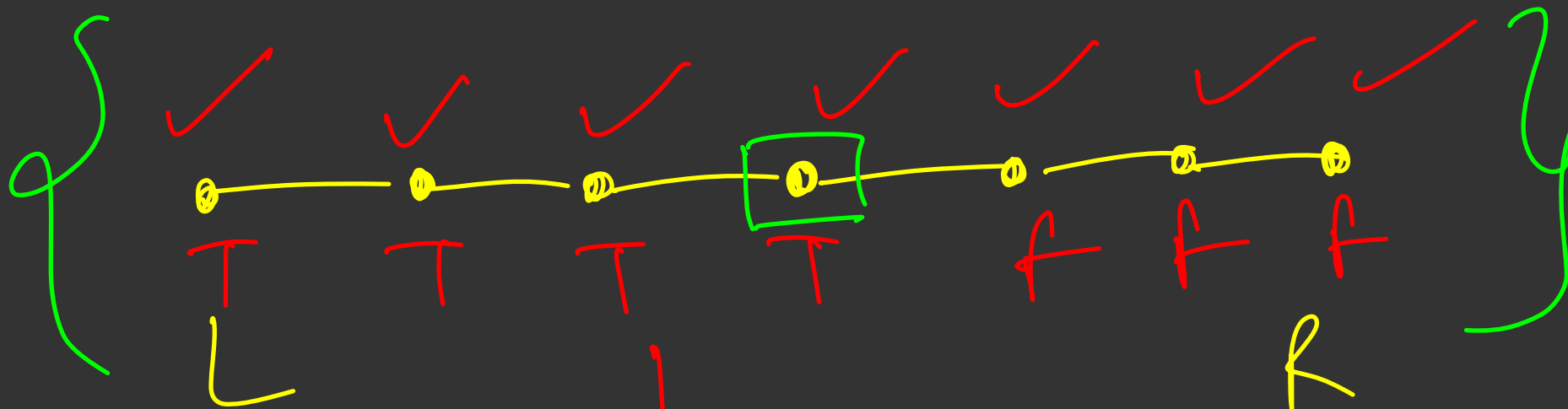


# Binary Searching with decimals

## Advanced Binary Search 2



$$f(x) = x^2$$

$$\underline{\underline{f(x) = 10.596}}$$

10.5 10.6 10.7 10.8 10.9 11.0 11.1



10.55




10.575




$$ans = 3.333333 \dots$$

Print any answer which is  
not farther than actual answer  
by  $10^{-7}$

10. 7 9 8 6 2 2 3 | 4



10. 7 9 8 6 2 2 3 | 5



find out  $\text{sqrt}(x)$  such that  
the answer is correct upto 10  
decimal places

split(x)

for integer



if  $(mid * mid \leq x)$   $\leq x \leq 100$





# Integer Binary Search

$$\boxed{10^{-10}}$$


$$\boxed{0 \longrightarrow 10^{16}}$$

$$\boxed{\log_2 10^{16}}$$

Total space

$$= \boxed{\text{original} \cdot \frac{1}{\text{precision}}}$$

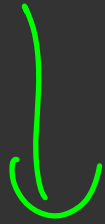
$$0 \text{ --- } 1 \quad 10^{10}$$


$$1 \text{ --- } 2 \quad 10^{10}$$


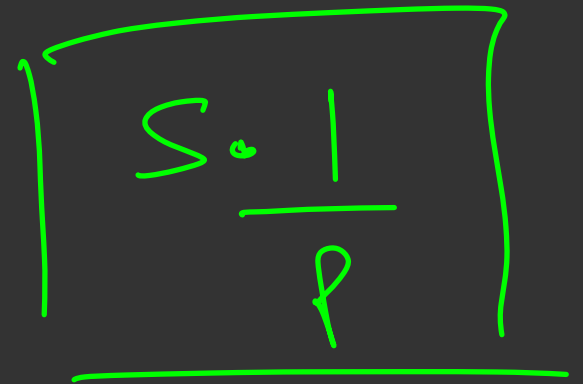
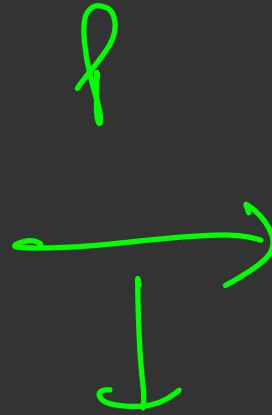
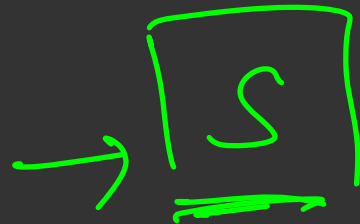
Integer



Decimal



Total search space



$$\log_2(S)$$

$$\begin{aligned} &10^{-10} \\ &\text{or} \\ &10^{-9} \end{aligned}$$

$$\log_2\left(S \cdot \frac{1}{p}\right)$$

Integer

0 to  $10^9$   
                    

$$\log_2 10^9$$

11  
⌋

30

$10^{-6}$   
\_\_\_\_\_>

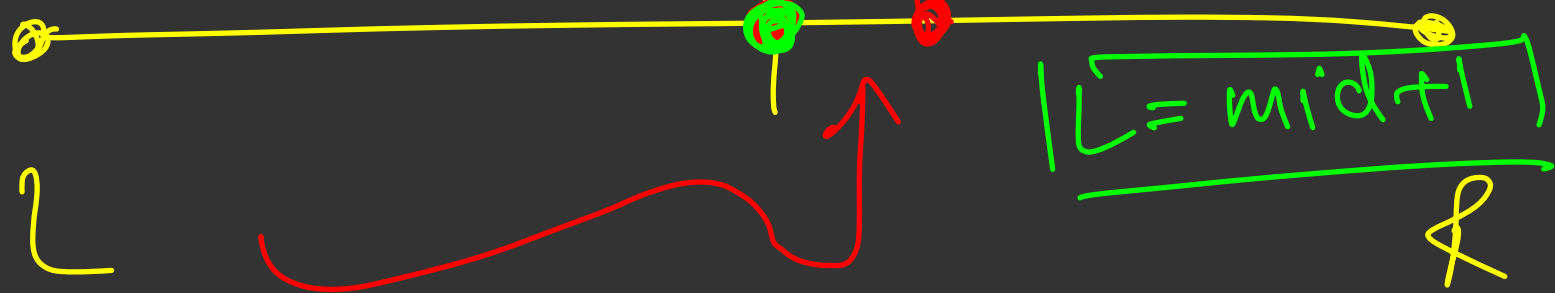
Decimal

$$10^9 \cdot \frac{1}{10^{-6}}$$

$$= 10^{15}$$

$$= 50$$

1 2 3 4 5 6 7 8 9 10 11 12 13



T T T T T T T f f f f

$mid + 10^{-6}$

$while (l < r)$

$while (l \leq r)$

while( $L \leq R$ )

R 10. 5000000 | 0 <sup>1</sup> 0 | 00 1

→ L 10. 5000000 | 0 <sup>2</sup> 0 | 00 2

$10^{-6}$

while (  $R - L > 10^{-6}$  )

$$\boxed{L, R}$$

$$\boxed{10^{-6}}$$

$\odot$        $\odot$        $\odot$        $\odot$        $\odot$   
 1      2      3      4      5

$$\boxed{L = R}$$

$$\boxed{L = R}$$

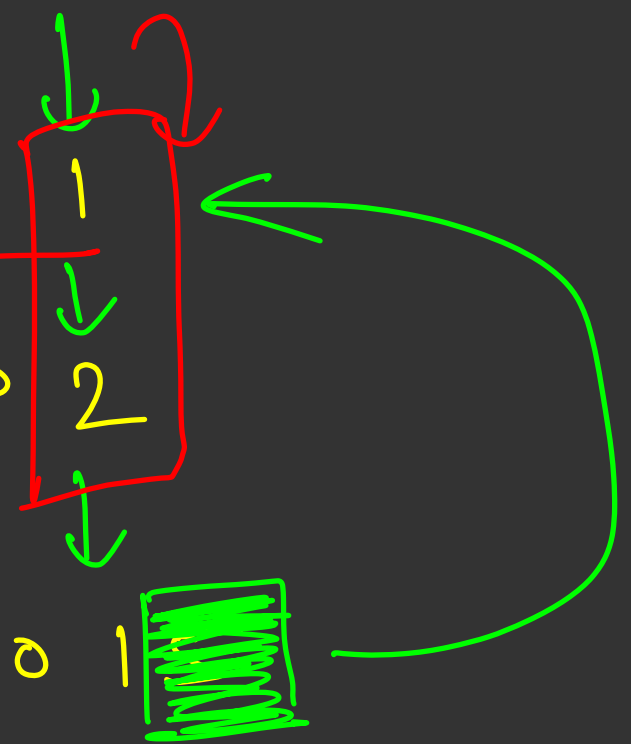
$\odot$   
 0.00001

$\odot$   
 0.50002

$L = 10.000000$

$R = 10.000000$

$mid = 10.000000$



# Binary Search on Decimals

Find a real number between L and R following certain properties.

- How many iterations?
- Precision issues?
- Ways to tackle

$$\log_2\left(S \cdot \frac{1}{p}\right)$$



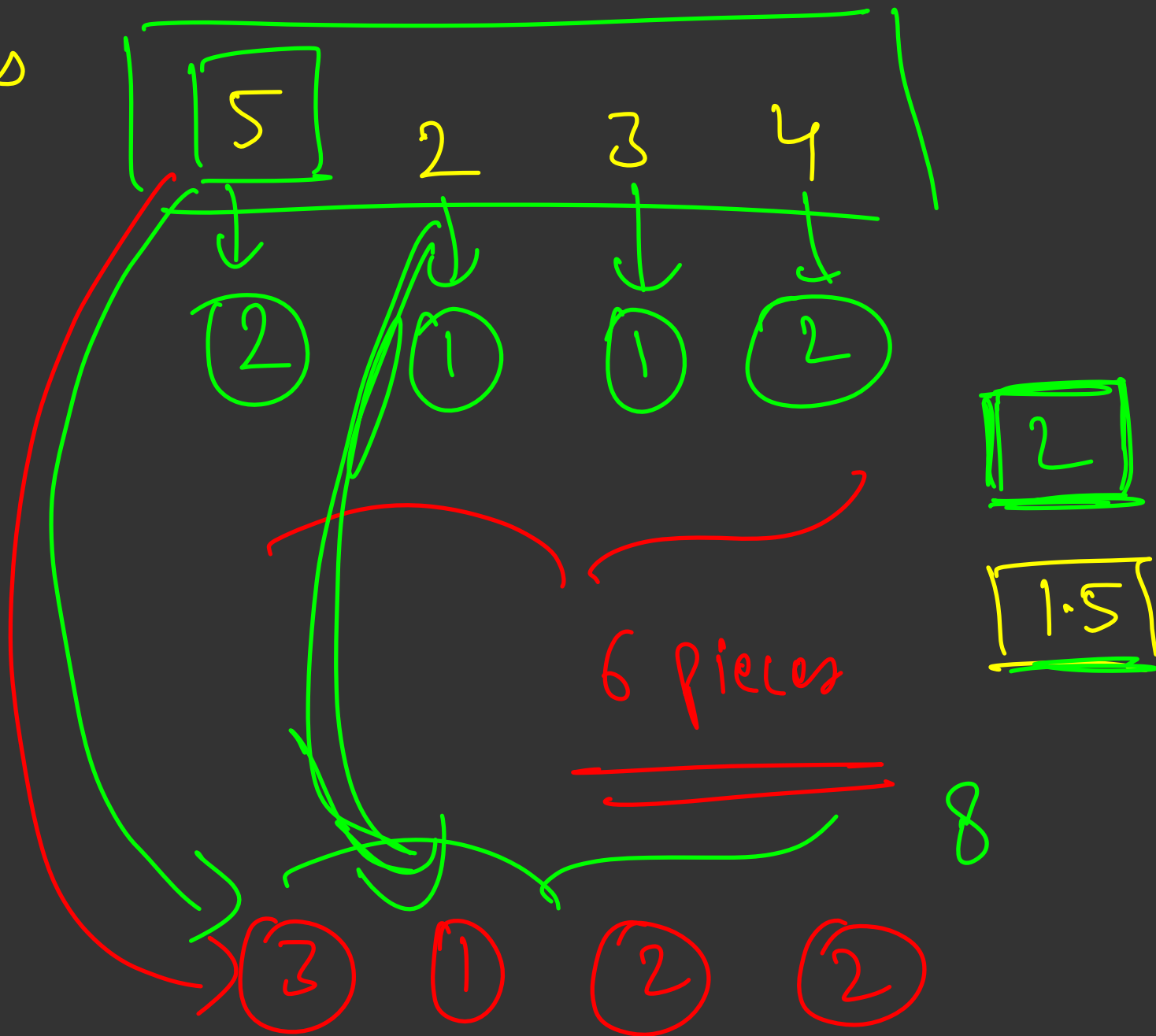
# Problem 1: Ropes

n ropes



k pieces

$$\frac{2}{1.5}$$

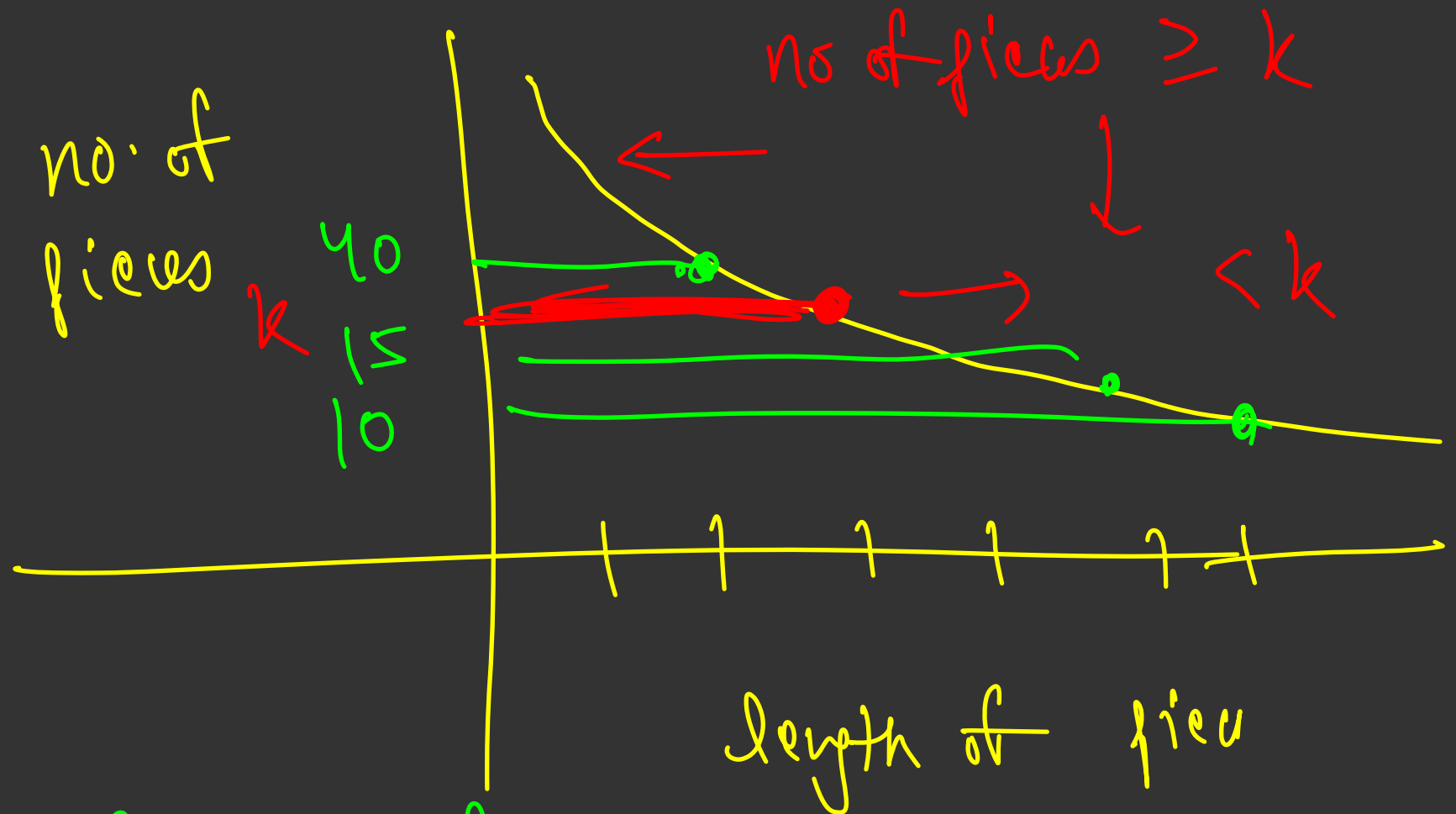


Obs. ①

reducing the length of piece

$\Rightarrow$  no. of total pieces that

we can get increases  $\uparrow$



max length of  
piece such that you can get  $k$   
pieces

$f(x)$   $\begin{cases} T & \text{if no. of pieces} \geq k \\ f & \text{if no. of pieces} < k \end{cases}$

$\boxed{==k}$

mid



L

if (  $f(\text{mid}) == T$  ) {

$\text{ans} = \text{max}(\text{ans}, \text{mid})$

$\text{l} = \text{mid} + 1$

} else {

$\text{r} = \text{mid} - 1$

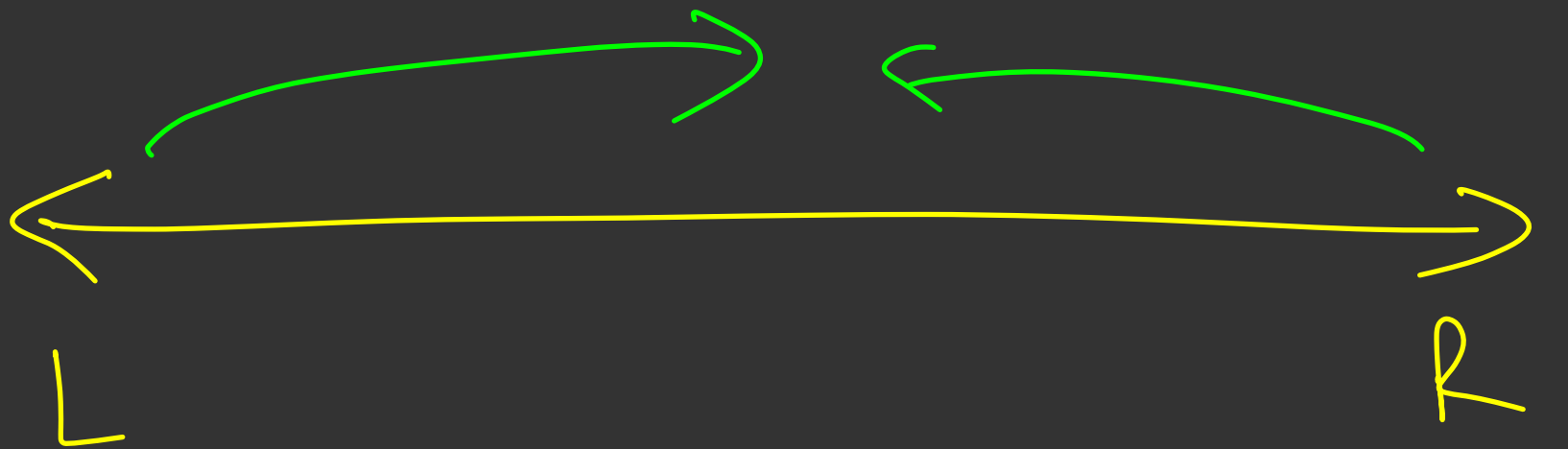
}

$f(x)$   $\rightarrow$   $O(n)$   $\cdot O(\log_2(\text{search}))$   
int sum = 0

for (int i = 0; i < n; i++)

sum +=  $\lfloor \text{arr}[i] / x \rfloor$

$f(x) = \text{sum} \geq k$



T T T T T T T f f f f f f f

LR



$$\boxed{0 \text{ to } 10^{12}} \quad \boxed{p = 10^{-12}}$$

$$0, 0.5001, 0.0002, \dots, \underline{10.0000}$$

$$\underline{\underline{10^4}}$$

$$\boxed{0, 1, 2, \dots, 10^5}$$

```

bool checker(double x, vector<int>& arr, int k){
    long long total = 0;
    for(auto i : arr){
        total += i / x;
        if(total >= k){
            return true;
        }
    }
    return false;
}

void solve(){
    int n, k;
    cin >> n >> k;
    vector<int> arr(n);
    for(int i = 0; i < n; i++){
        cin >> arr[i];
    }
    double left = 0, right = 1e7, error = 1e-7;
    double ans = 0;
    while(right - left > error){
        double mid = (left + right) / 2;
        if(checker(mid, arr, k)){
            ans = max(ans, mid);
            left = mid + error;
        }else{
            right = mid - error;
        }
    }
    cout << setprecision(7) << fixed << ans << endl;
}

```

$10^{-6}$

0.001

0.005  
0.004

$10^{-6}$

$10^{-7}$

Any problems  
with this code?

float | int

= int      floored value

$$\left[ x_1/y + x_2/y \quad \dots \quad x_n/y \right]$$

$$10.00001$$

$$10$$

Array

$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$
-------	-------	-------	-------	-------	-------

$$\left[ \begin{aligned} &a_1/x + a_2/x + a_3/x + a_4/x \\ &\quad + a_5/x + a_6/x \end{aligned} \right]$$

$$= 10$$

$$\rightarrow 8$$

$$\text{for } x = 1$$

$$x = \underline{\underline{1.1}}$$





$$\theta_{\text{max}} = 10^{-3}$$

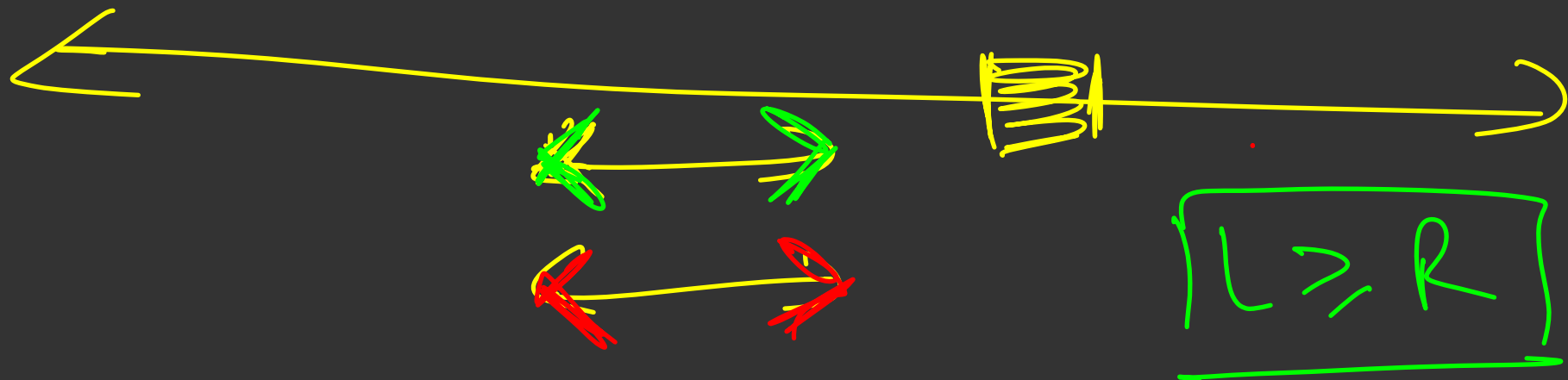
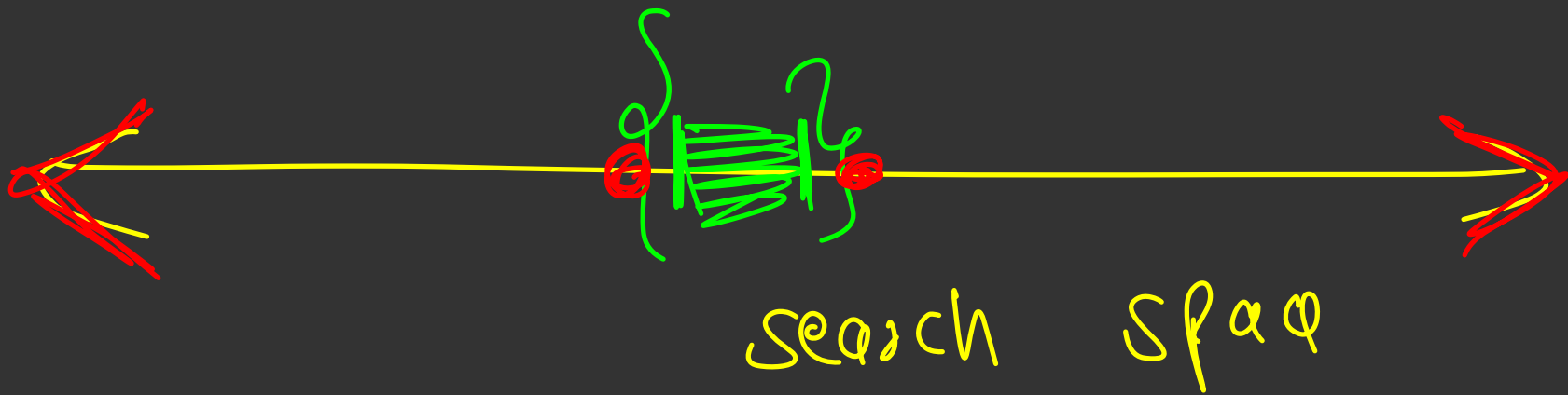
$$\alpha = 0.001$$

$$\gamma = 0.002$$

$$\underline{\underline{10^{-6}}}$$

$$\underline{\underline{10^{-8}}}$$

1 2 3 4 5 6 7 8 9 10



$0.0000001$   
 $0.0000002$  }  $10^{-6}$



$$\left\lfloor \frac{L+R}{2} \right\rfloor$$

float

10.000000000000001

10.000000000000002

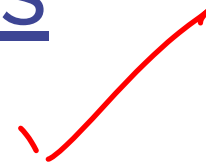
$$L > R$$

false

$$L > R$$

True

## Problem 2: Packing Rectangles



```

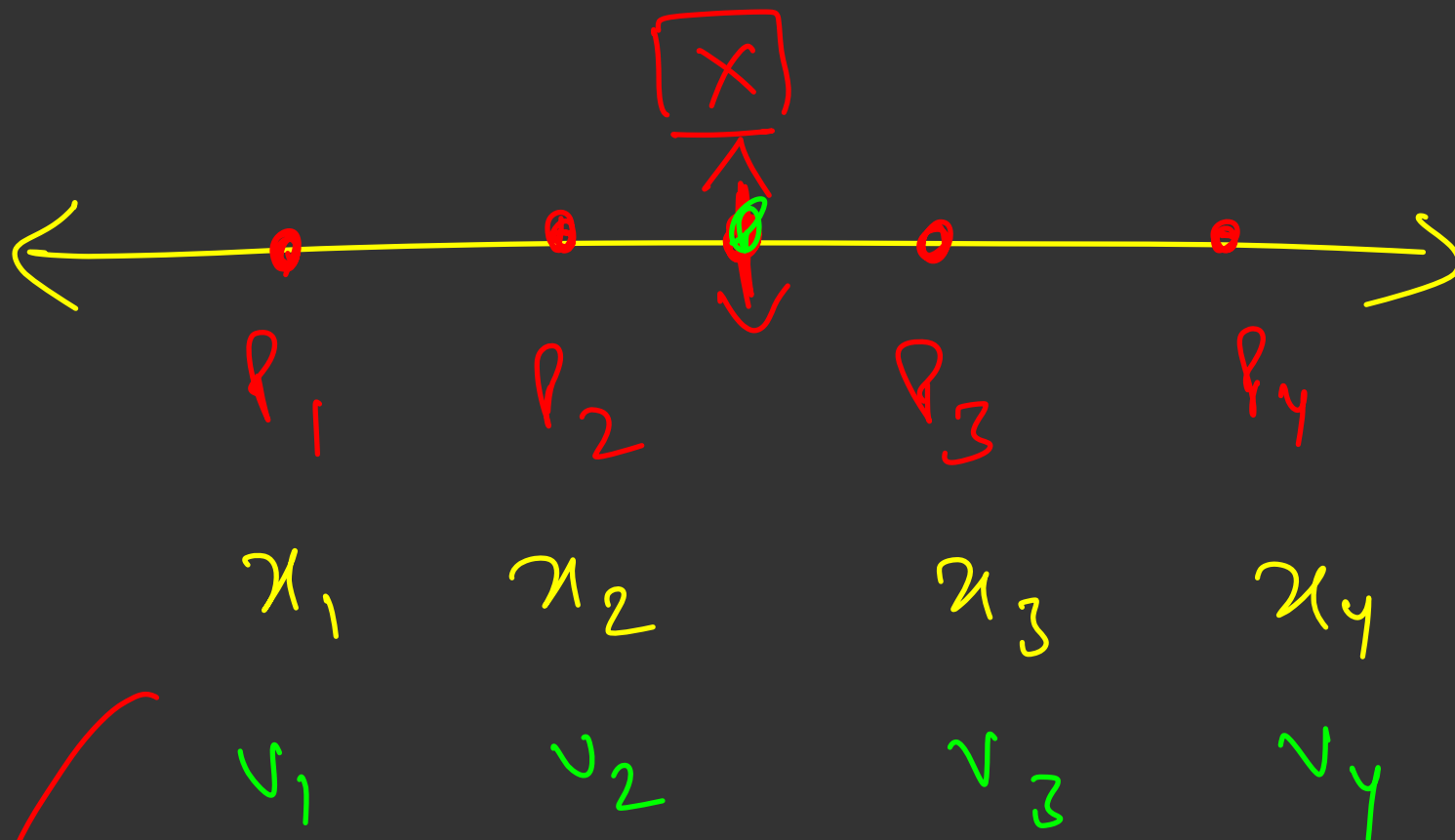
void solve(){
    long long w, h, n;
    cin >> w >> h >> n;
    long long left = 1, right = 1e18;
    long long ans = 1e18;
    while(left <= right){
        long long mid = (left + right) / 2;
        long long rows = mid / h;
        long long cols = mid / w;
        if(cols > 0 && rows >= (n + cols - 1) / cols){
            ans = min(ans, mid);
            right = mid - 1;
        }else{
            left = mid + 1;
        }
    }
    cout << ans << endl;
}

```

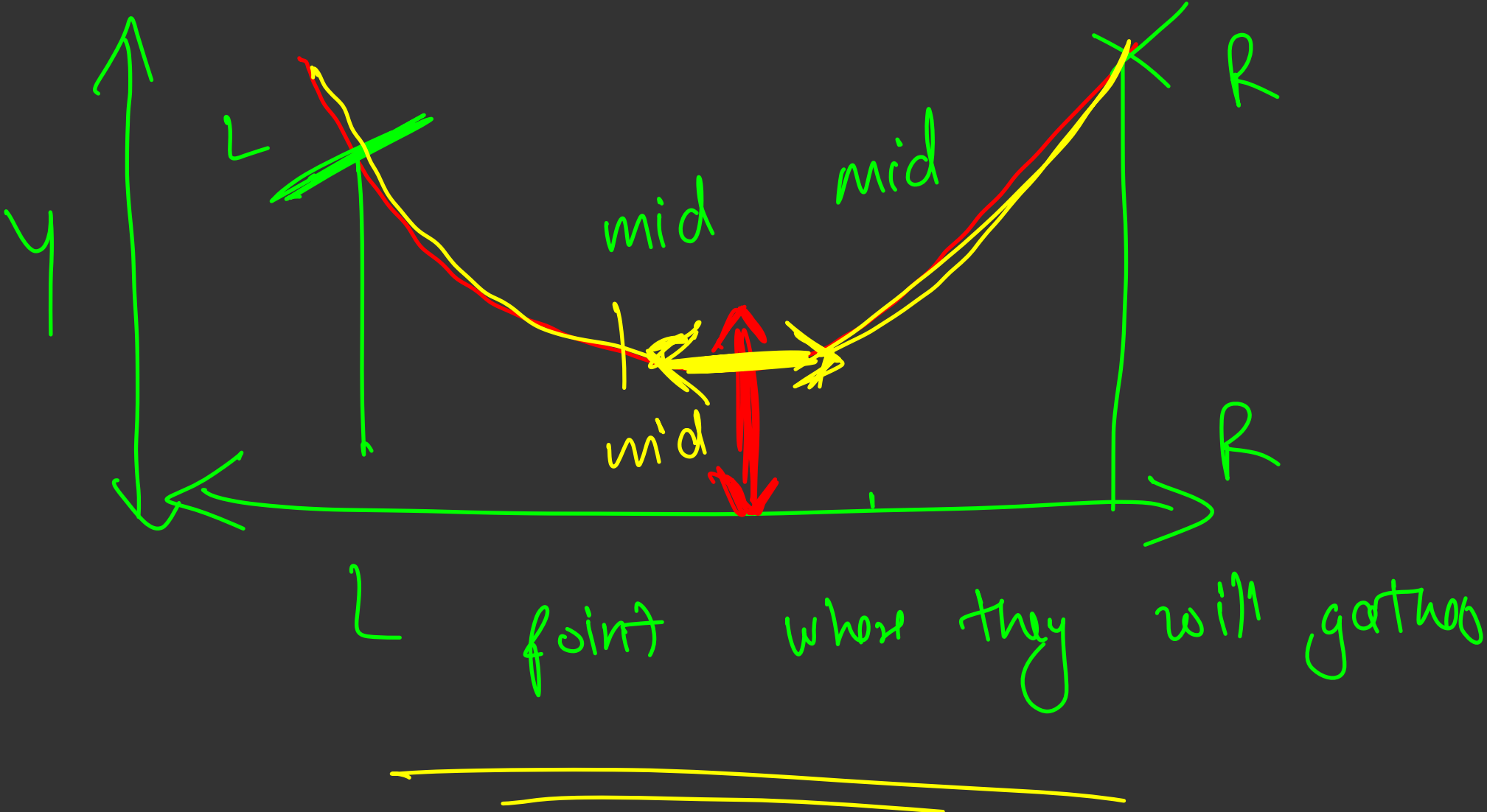
## Problem 2

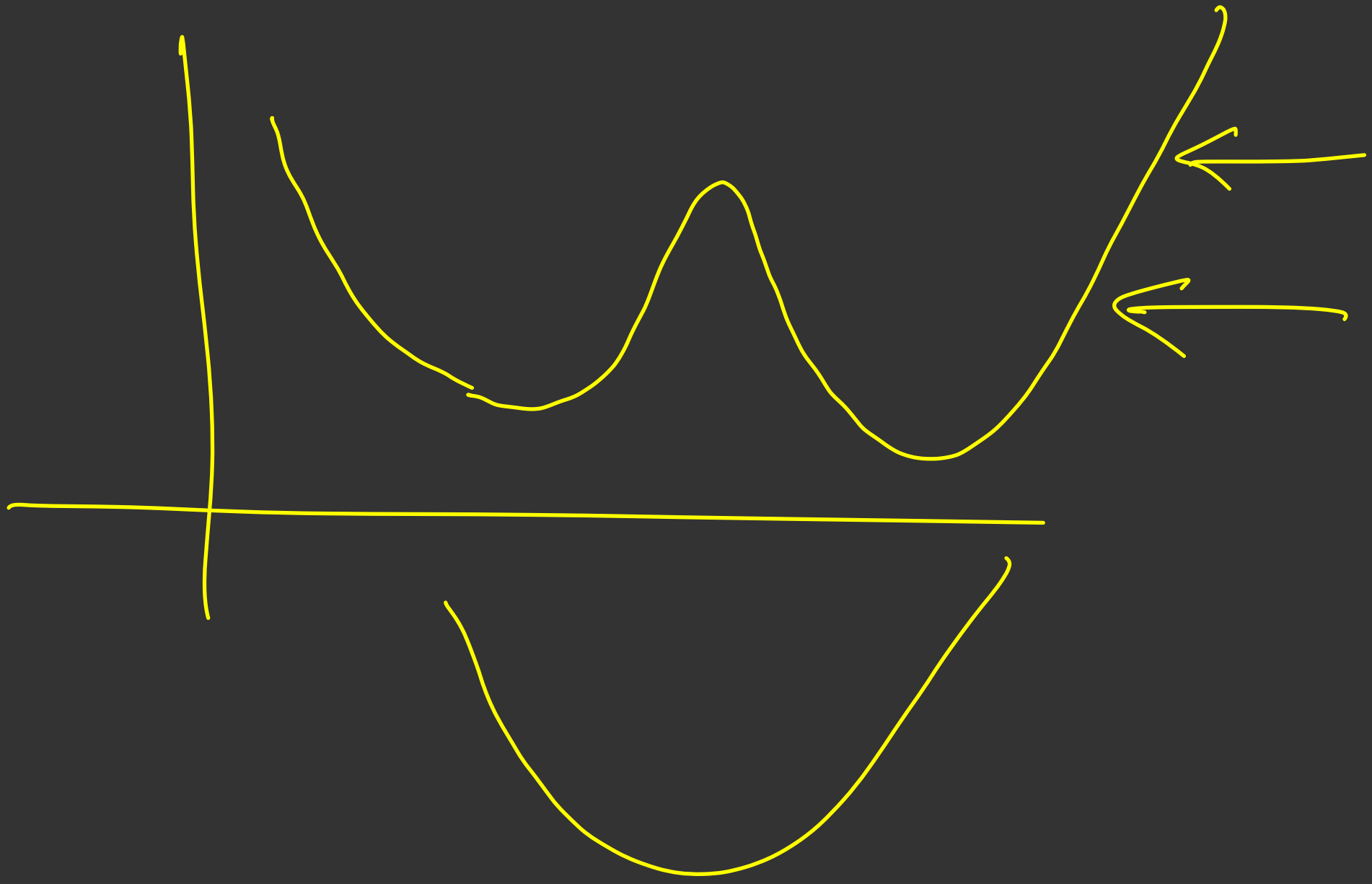
## Solution Code

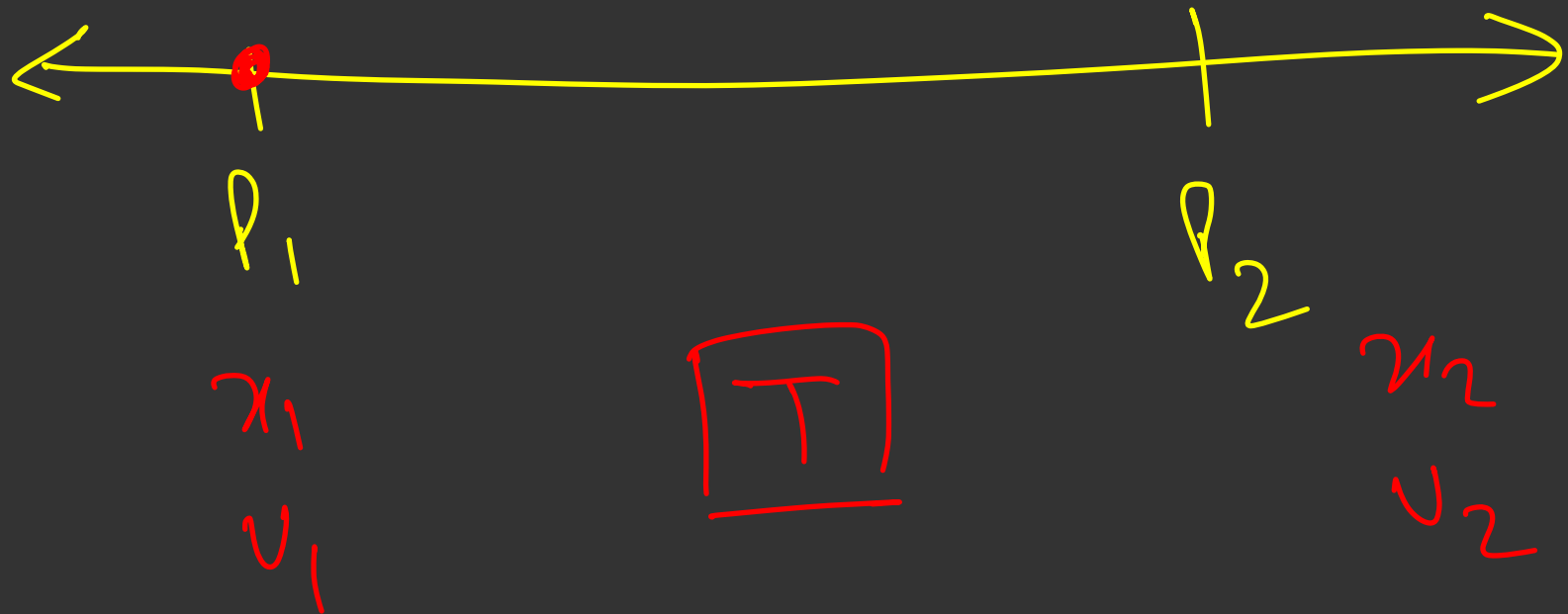
## Problem 3: Get Together



$m_{an}$ 
 $\left( \frac{|x-x_1|}{v_1} \right)^3$ 
 $\frac{|x-x_2|}{v_2}^4$ 
 $\frac{|x-x_3|}{v_3}^5$ 
 $\frac{|x-x_4|}{v_4}^2$

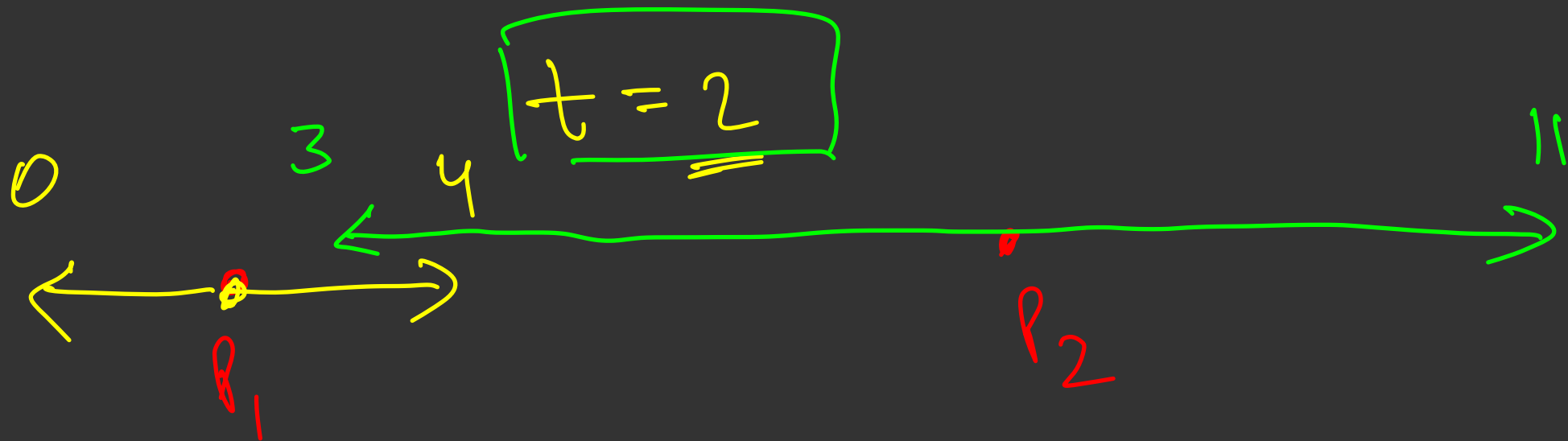






$$\left[ \underline{x_1 - v_1 t}, \quad \underline{x_1 + v_1 t} \right]$$





$$\underline{x_1 = 2}$$

$$v_1 = 1$$

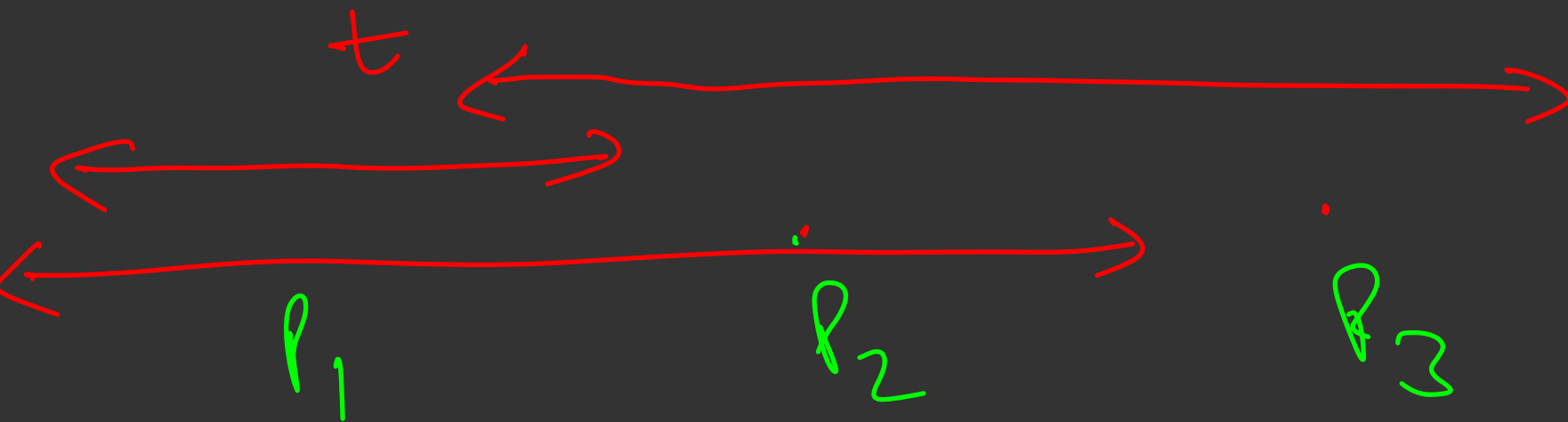
$$\underline{(0, 4)}$$

$$x_2 = 7$$

$$v_2 = \underline{2}$$

$$\left( \underline{7 - 2 \cdot 2}, 7 + 2 \cdot 2 \right)$$

$$\left[ 3, 11 \right)$$



$x_1$

$x_2$

$x_3$

$v_1$

$v_2$

$v_3$

$f(x)$   $\begin{cases} T \\ F \end{cases}$

if points can meet in  
time  $x$

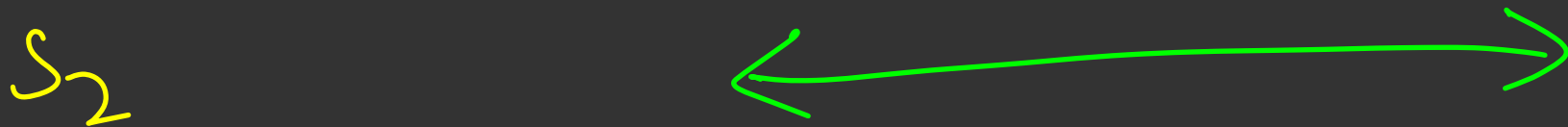
$O(\omega)$

$f(x)$

f f f f f T T T T T T T

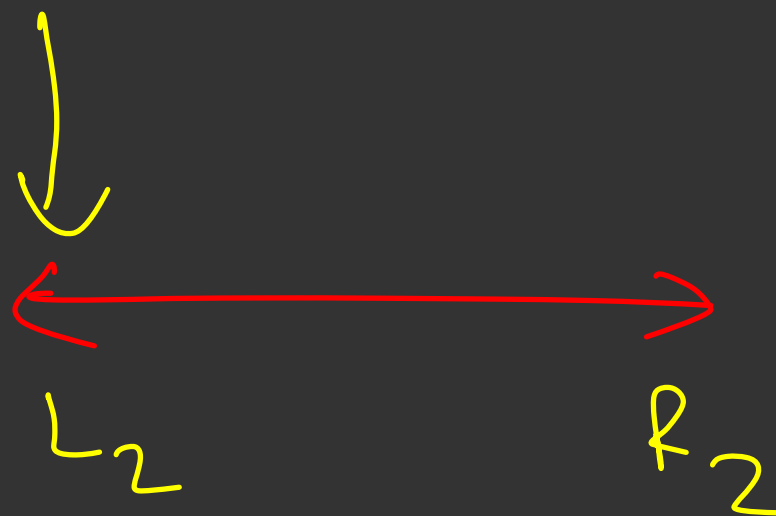
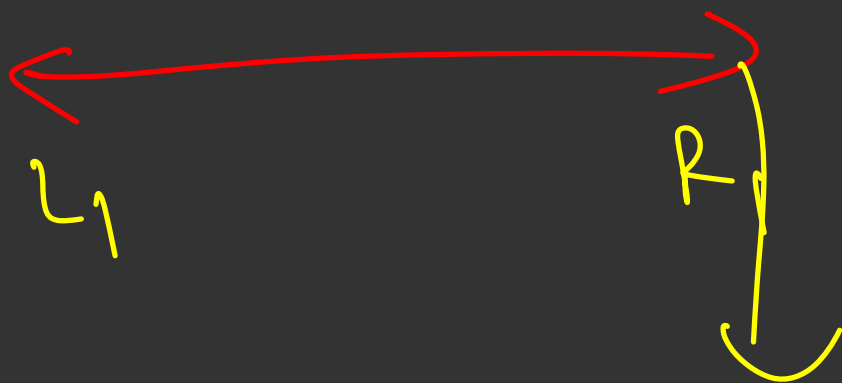


find out the intersection of  
 $n$  segments





$$\left[ \max(L_1, L_2), \min(R_1, R_2) \right]$$



$\max(L_1, L_2)$

$\min(R_1, R_2)$



```

bool checker(double t, vector<pair<double, double>>& people){
    pair<double, double> curr = {-1e18, 1e18};
    for(auto i : people){
        double x = i.first;
        double speed = i.second;
        pair<double, double> range = {x - speed * t, x + speed * t};
        curr = {max(curr.first, range.first), min(curr.second, range.second)};
        if(curr.first > curr.second)
            return false;
    }
    return true;
}

void solve(){
    int n;
    cin >> n;
    vector<pair<double, double>> people(n);
    for(int i = 0; i < n; i++){
        cin >> people[i].first >> people[i].second;
    }
    double left = 0, right = 1e9;
    for(int iteration = 0; iteration < 60; iteration++){
        double mid = (left + right) / 2;
        if(checker(mid, people)){
            right = mid;
        }else{
            left = mid;
        }
    }
    cout << setprecision(7) << fixed << right << endl;
}

```

$O(n)$

$f(t)$

$curr = (L_1, R_1)$   
 $range = (L_2, R_2)$

$O(\log(\text{search space}))$

Problem 3  
Solution Code

$(L_1 \leq R)$