Advanced Binary Search 2

- Priyansh Agarwal

FairWorkload Problem

Given an array of workloads, split it among **K** workers, such that the maximum work that any worker has to do is minimised (can't change order of workloads).

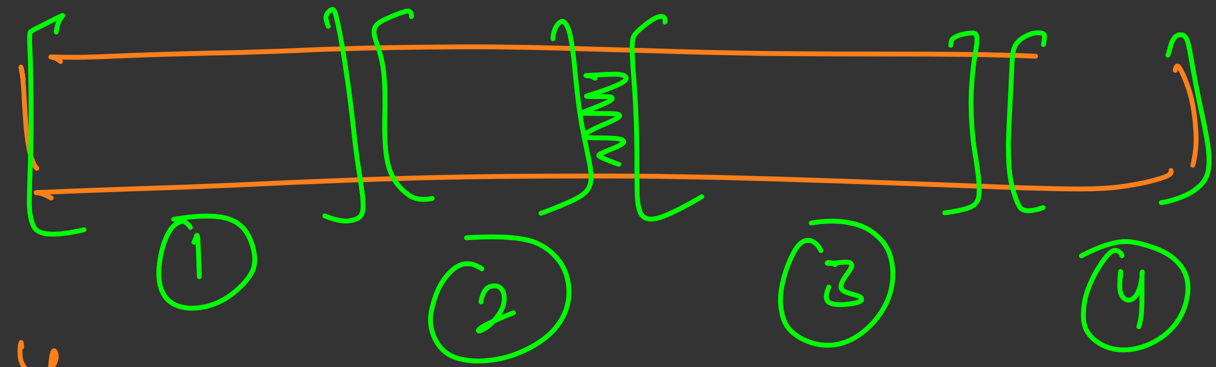
Eg. [10, 20, 30, 40, 50, 60, 70, 80, 90]

$K = 3$

- Solution : 10 20 30 40 50 | 60 70 | 80 90

  First worker - 150, Second worker - 130, Third worker - 170

Is it possible to partition workload in a way that the highest workload of any worker is less than 170?



$k = 4$

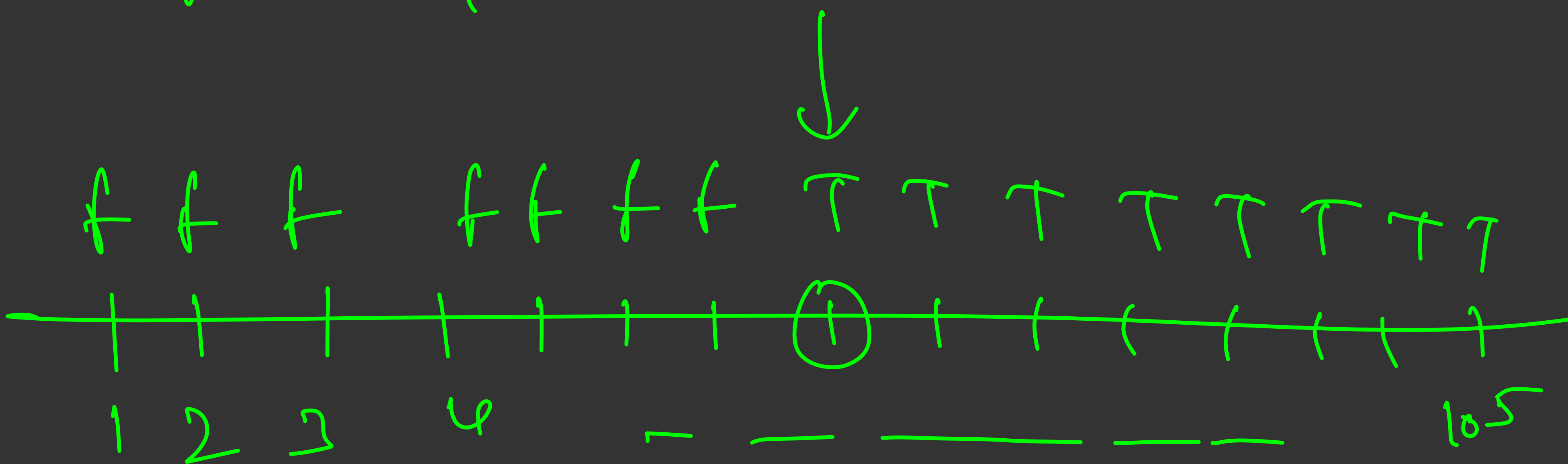
if I can split the array into k part such that every worker gets $\le X$ workload

Can I also split such that every worker gets $\le X + 5$

Predicate function $f(x)$

= True if array can be split into k segments such that every segment's sum $\leq x$

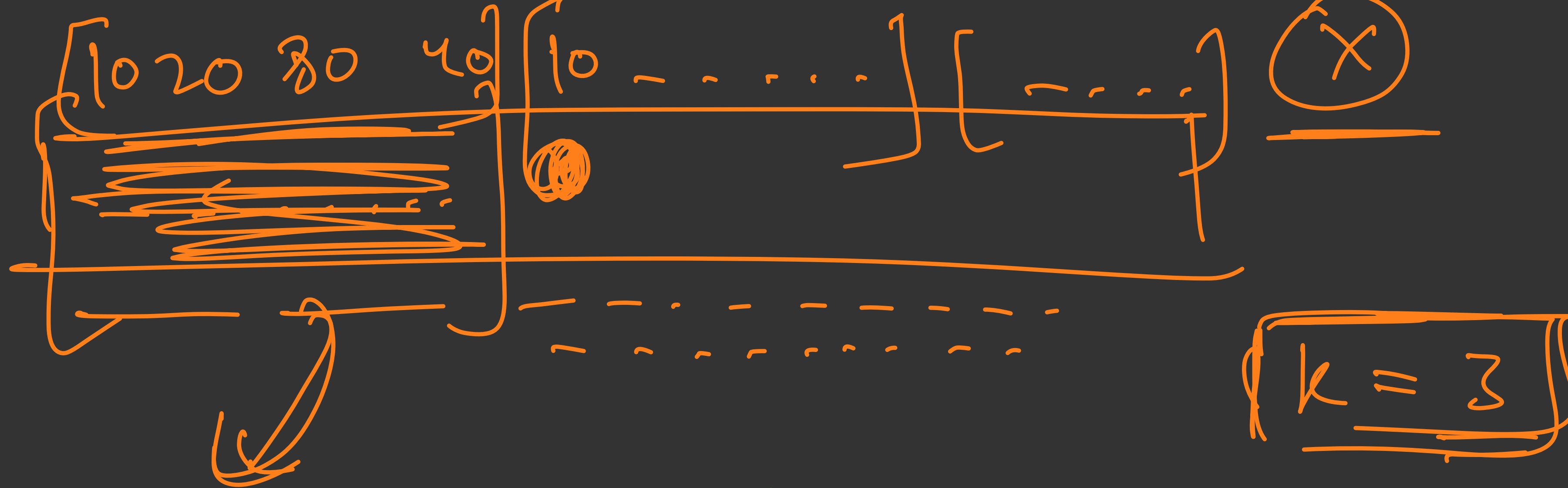
false o/w



Binary Search on a Search space

$T.C = O(\log(S.S) \cdot$ Time to check every candidate ans$)$

$f(x)$ — T / F: can we split array into k segments for which every segment sum $\leq x$



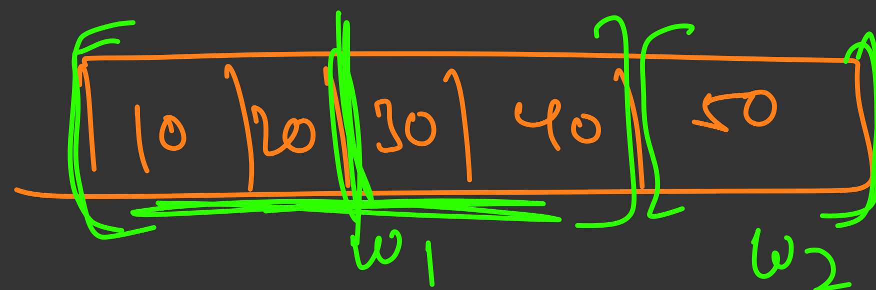
$k = 3$

w_1

5

$x = 100$

$x = 101$



Define search space

$\rightarrow N \le 10^5$
$\rightarrow w_i \le 10^9$ } $\underline{10^{14}}$

[$\underline{\boxed{\text{Lower}}}$] [$\underline{\text{High}}$]

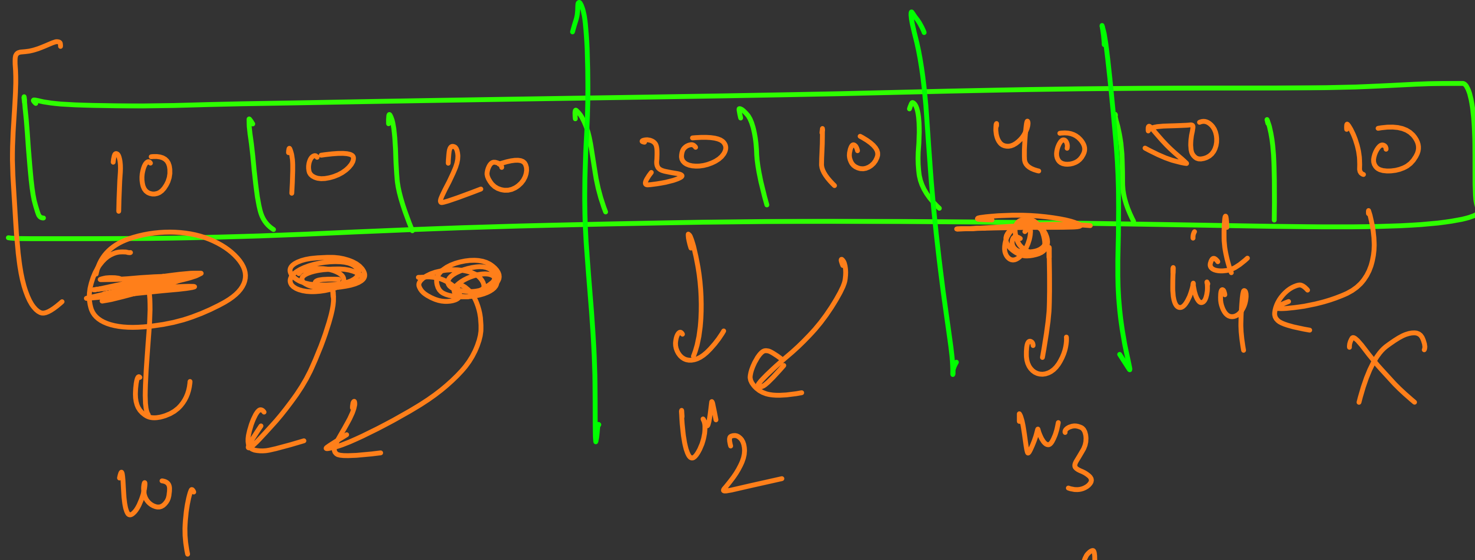
60

[max element in the array] [sum of all elements]

$\underline{\underline{0}}$

10^{14} $\boxed{10^{18}}$ 10^{15}

$\log(S.S) \cdot O(n)$ 46 60 50



$X = 60$

$f(x)$

S.S $\longrightarrow$ 10^{18}

T.C $\longrightarrow$ $O(n \cdot \log(10^{18}))$

f f f f f f f f f [f f f f T T T T T T]

L

mid

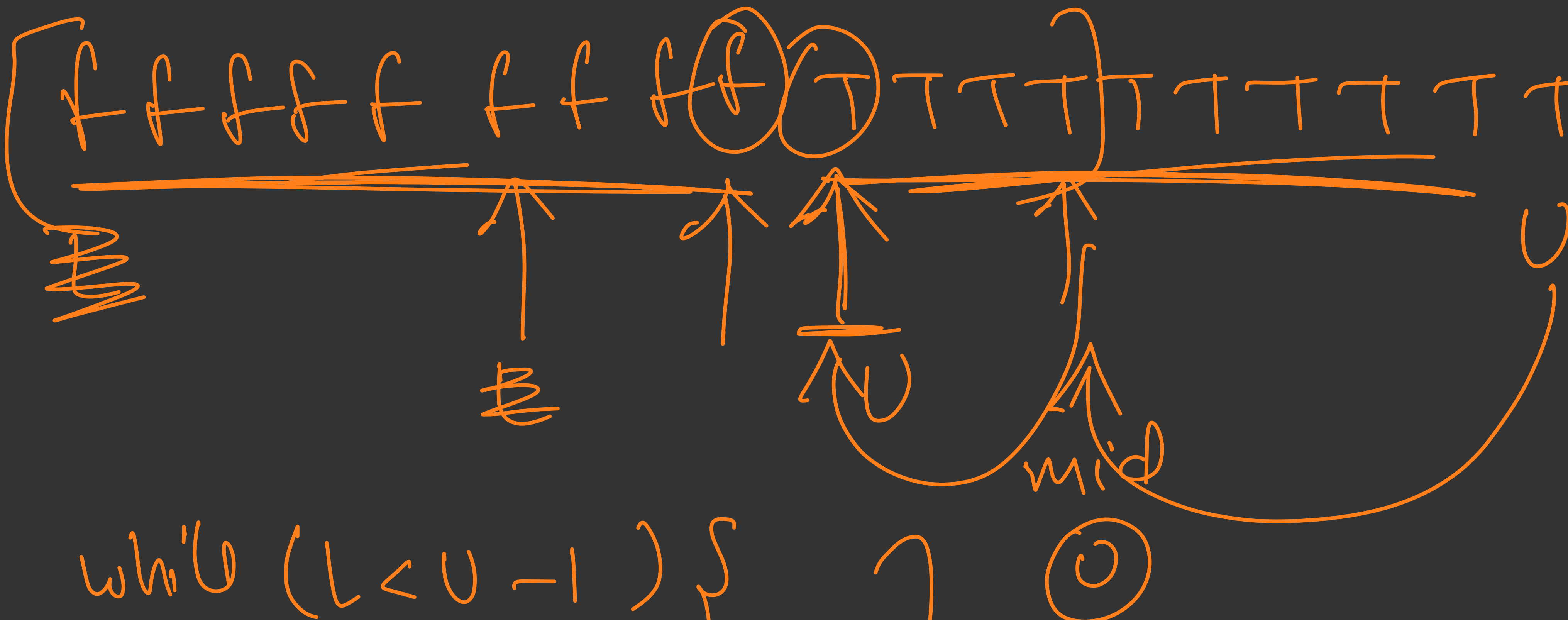
L = mid + 1

mid

U

U = mid - 1

Ans =




```
while (l < u - 1) {
    mid = (l + u) / 2
    if (f(mid))
        u = mid
    else
        l = mid
}
```

Sqrt(X)

1 10^9 $\boxed{X}$

Given a number X ($1 <= X <= 10^{18}$), find the biggest number Y such that $Y^2 <= X$

Eg:

X = 10000, Y = 100

X = 1024, Y = 32

X = 26, Y = 5

$Y^2 \leq X$

$y_4^2 \leq X \Rightarrow y_3^2 \leq X \Rightarrow y_2^2 \leq x$

$y_1 \leq y_2 \leq y_3 \leq y_4 \leq y_5 \leq y_6 \leq - - - - - - - - y_{1000}$

$$f(y) \begin{cases} \text{True} & \text{if } y^2 \le X \\ \text{false} & \text{o/w} \end{cases}$$

L U

T T T T T T T T T T T T | f f f f f f f f f

LB → , UB →

while (LB ≤ UB) {

mid = $\frac{LB+UB}{2}$

if ((mid x mid ≤ X))

ans = mid

LB = mid +1

else

UB = mid −1

}

mid x mid ≤ x

mid ≤ $\frac{x}{mid}$

Packing Rectangles

n rectangles $(w \times h)$

100

101
102

$f(x) \longrightarrow$ True if all rectangles can be fit into a square of side x

F F F F F F F T T T T T T T T T T T T T T T T

Square of side x, how to check if n rectangles can fit into this square

$x = 5$

$w = 2$

$$\left\lfloor \frac{x}{w} \right\rfloor \times \left\lfloor \frac{x}{h} \right\rfloor \geq n$$

$1 \le n, h, w \le 10^9$

$\boxed{10^{18}}$

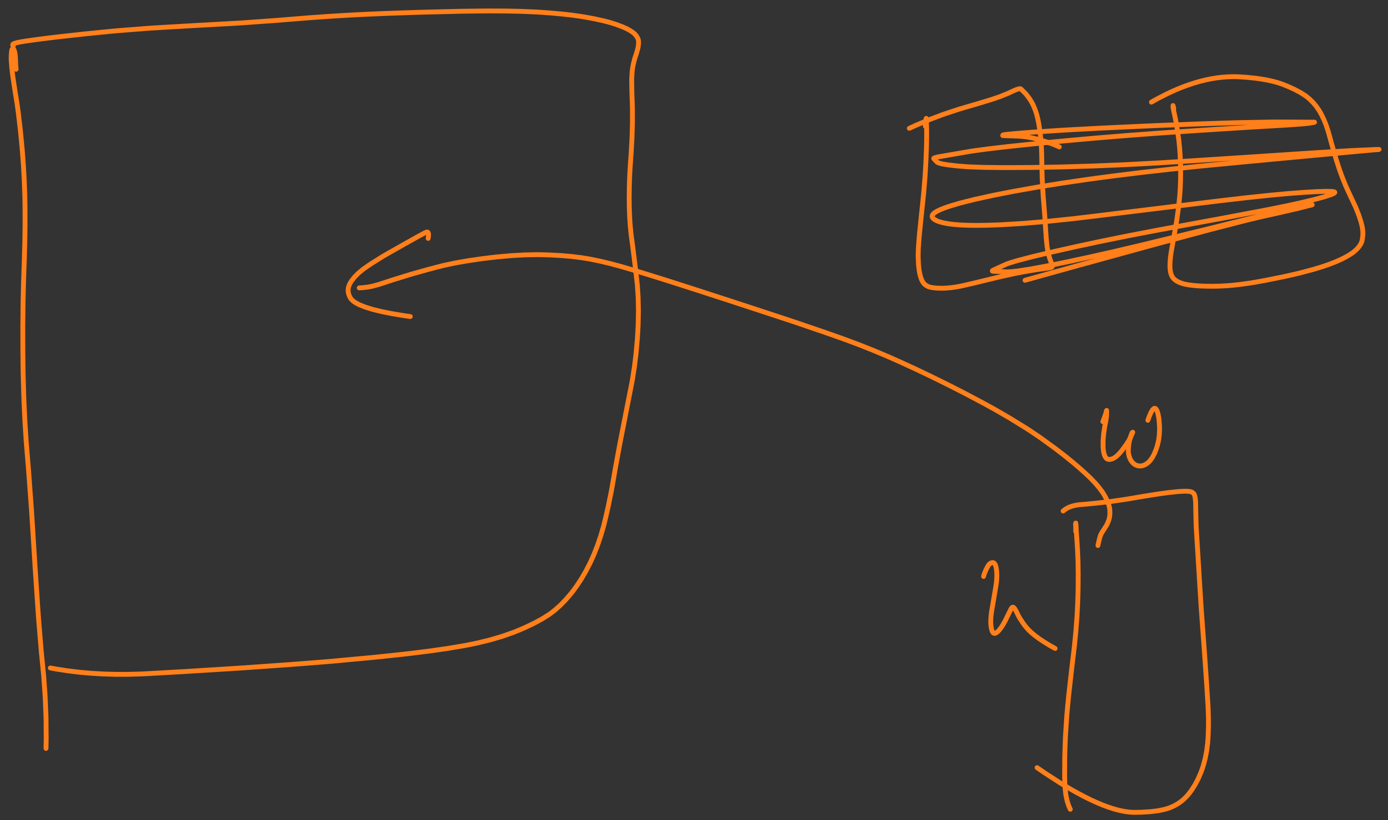
$n \to \underline{10^9}$ $w \to \underline{10^9}$

$\boxed{lb = 1}$ $\boxed{ub \ge 10^{18}}$

$f(x) \to \frac{x}{w} \times \frac{x}{h} \ge n$

$w = 1$, $h = 1$

10^{18} 10^{18} 10^{18}


w
h

$$\boxed{a \times b \geq c}$$

10^{18} 10^{18} 10^{18}

$$a \geq \left\lceil \frac{c}{b} \right\rceil$$

$a = 2$

$b = 3$

$c = 7$

$a \times b \geq c$ ✗

$$a \geq \left\lceil \frac{c}{b} \right\rceil$$

$$\geq \left\lceil \frac{7}{3} \right\rceil$$

$$\geq 3$$

$a \times b \geq c$ $\qquad$ a, b, c are decimals

$$\downarrow$$

$$\boxed{a} \geq \boxed{\frac{c}{b}}$$

$a \geq k + x$

$a \geq k + 1$

if

$\frac{c}{b} =$ some integer k

$+$ some fraction x

$0 \leq x < 1$

$$\frac{c}{b} = k + x$$

$a \geq c$

Solution Code

```cpp
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;
}
```

UB

UB

$\boxed{rows \times cols \geq n}$

$rows \geq \lceil n/c \rceil$

$rows = mid/h$

n/h

$$\left\lceil \frac{n}{c} \right\rceil = \frac{n+c-1}{c}$$