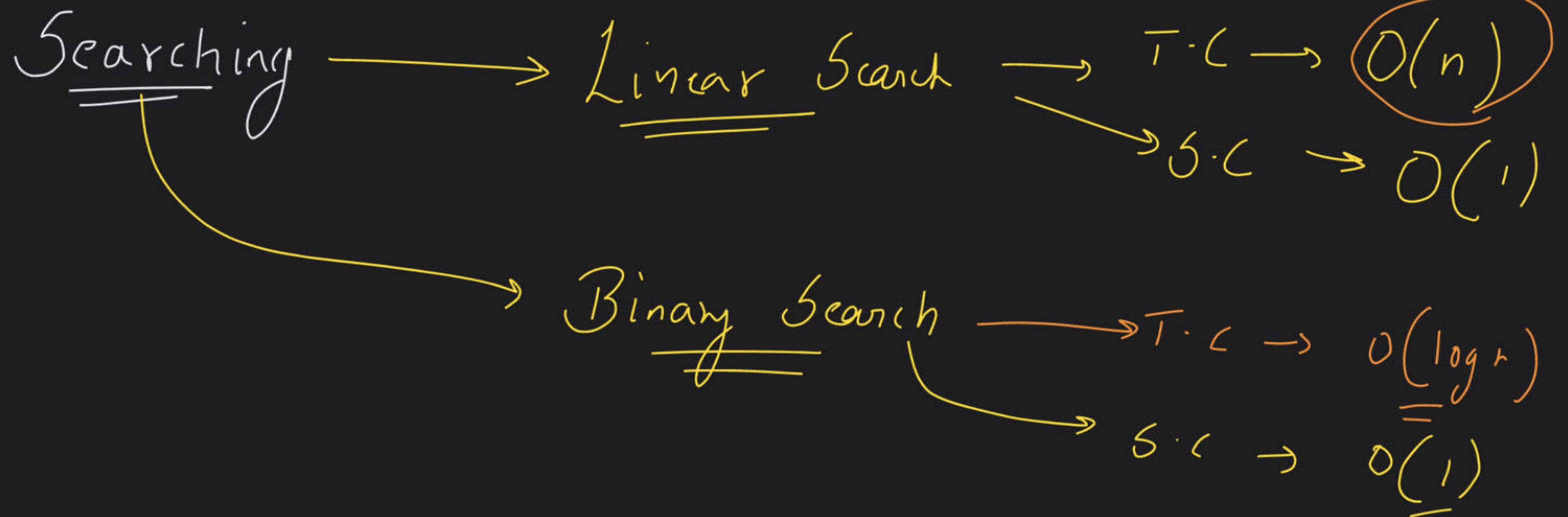


Searching and Sorting Class - 1

Special class

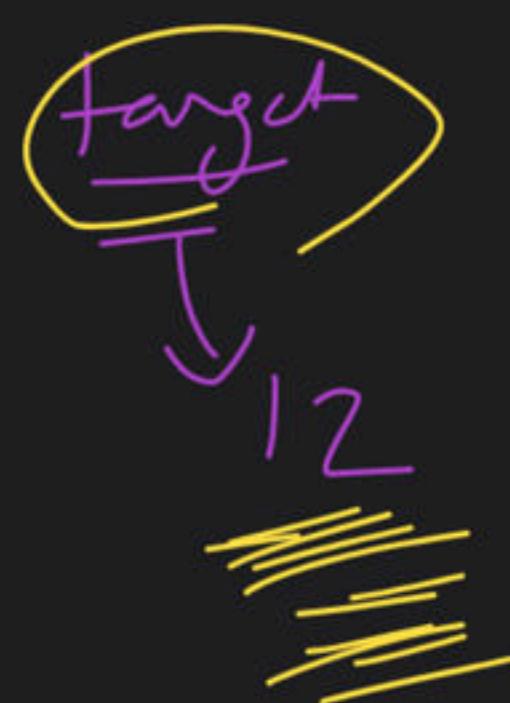
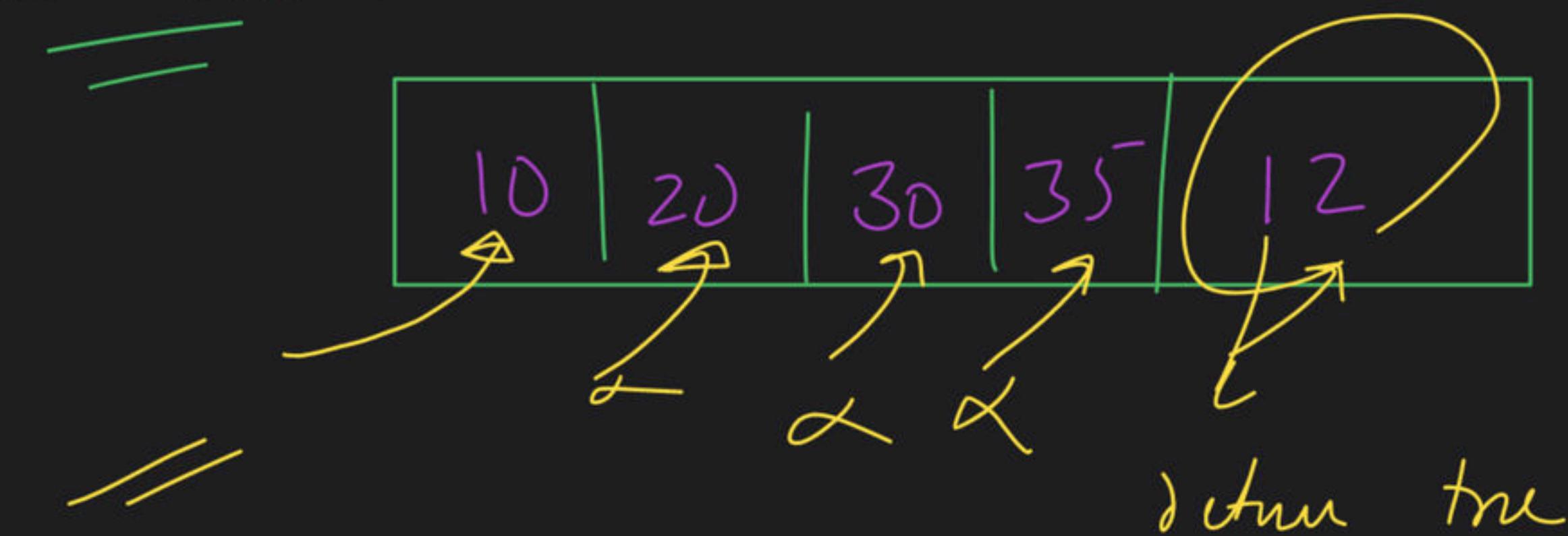
Revision





Binary Search \Rightarrow condition \rightarrow monotonic function

Linear Search





inc order
sorted array



$$s = 0$$

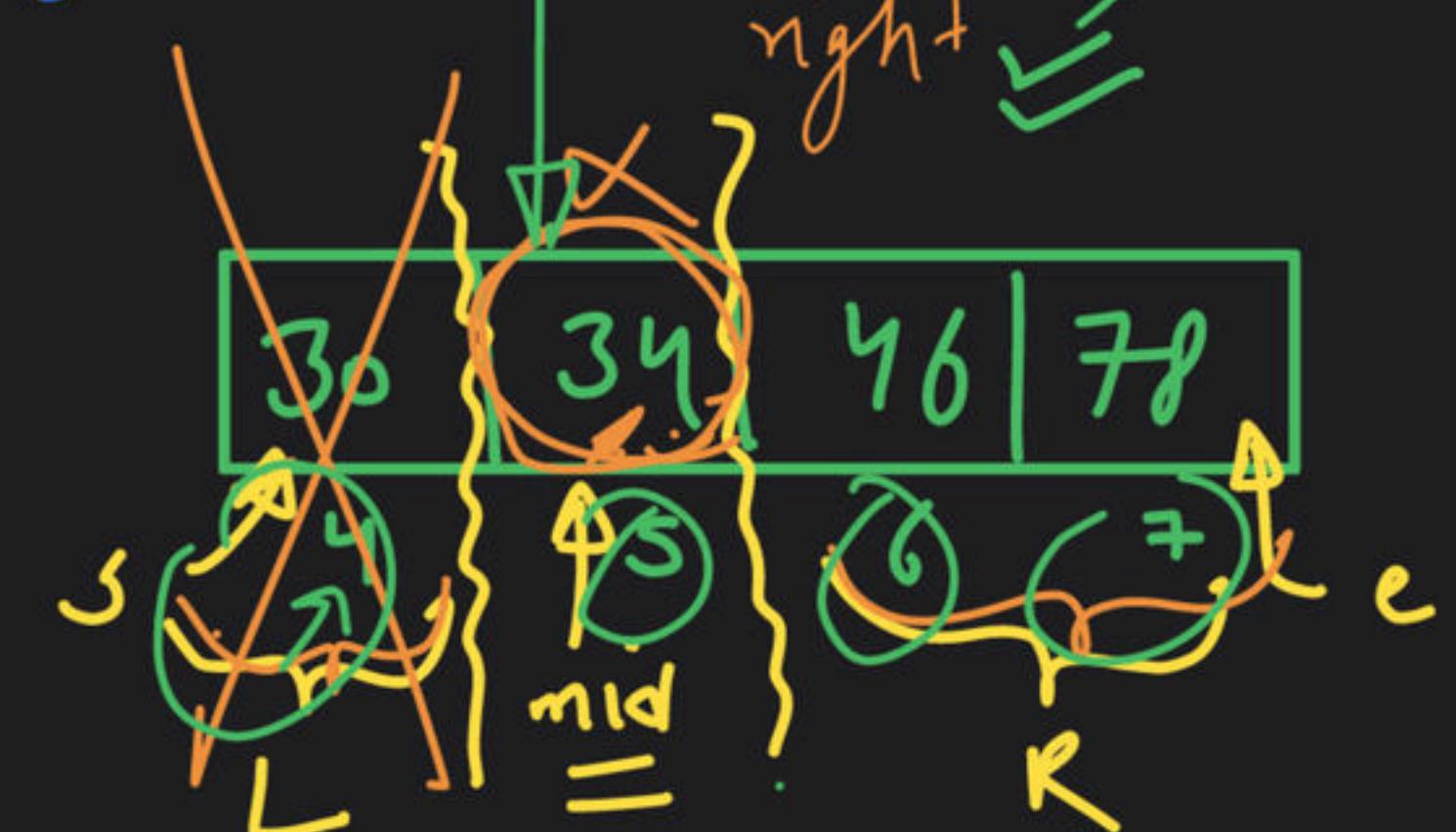
$$e = 7$$

$$\text{mid} = \frac{s+e}{2} = \frac{0+7}{2} = 3$$

$$s = 4$$

$$e = 7$$

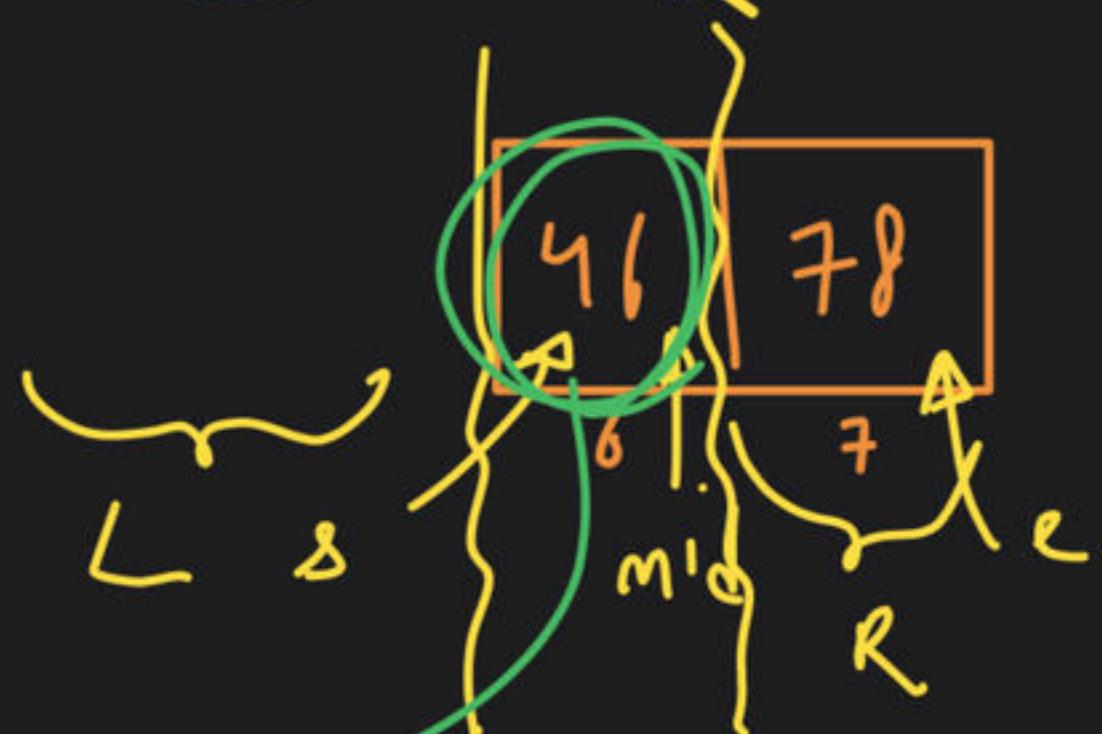
$$\text{mid} = \frac{s+e}{2} = \frac{4+7}{2} = 5$$



$$s = 1$$

$$e = 3$$

$$\text{mid} = \frac{s+e}{2} = \frac{1+3}{2} = 2$$

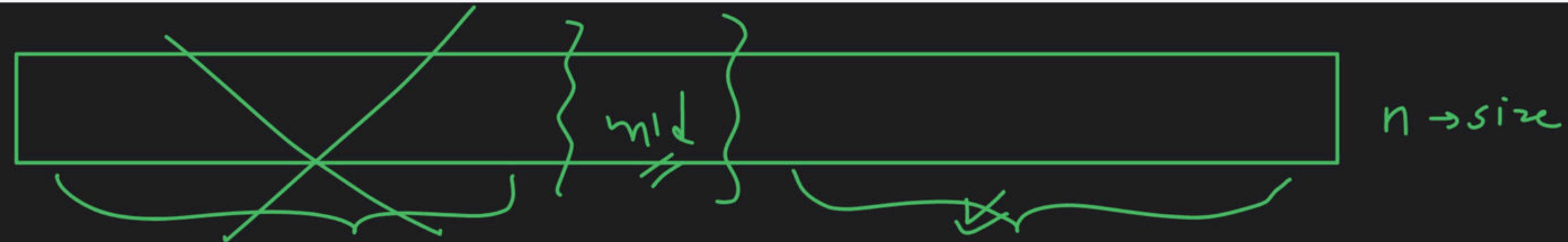


① / ② put

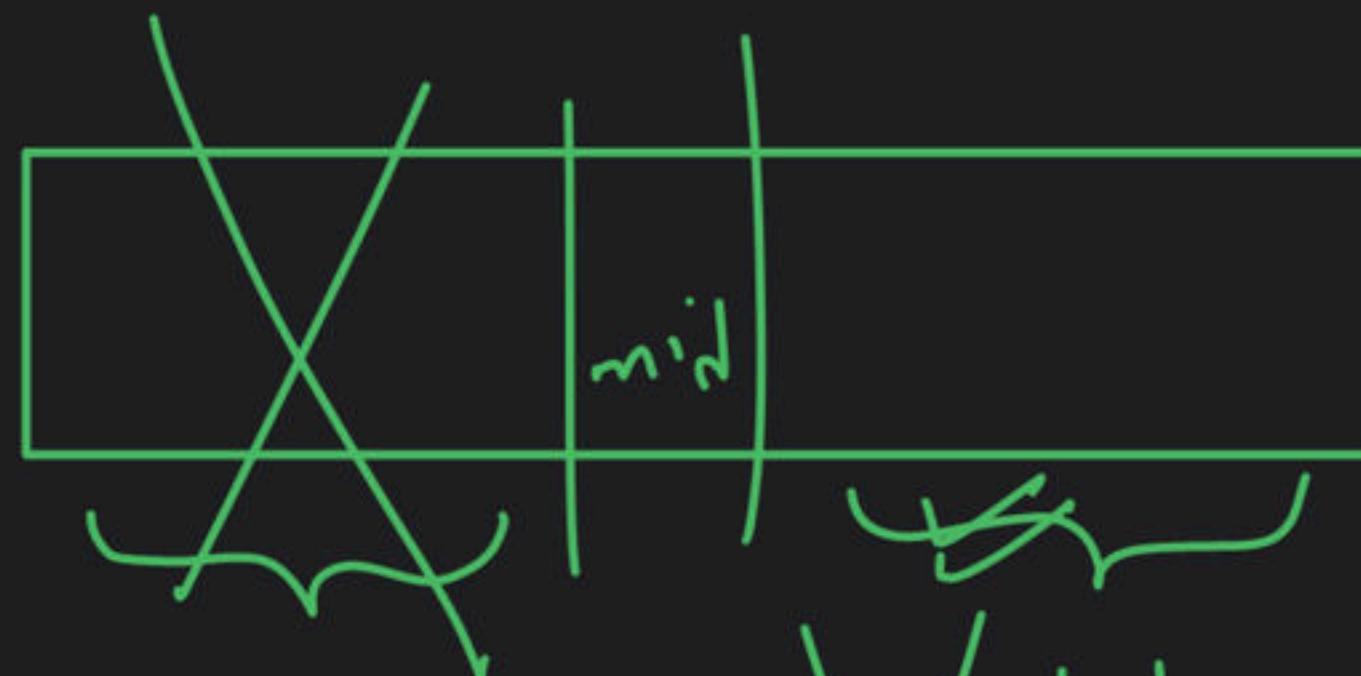
if (arr[mid] == target)

return true

return true


 $\frac{n}{1}$
 $\frac{n}{2}$
 $\frac{n}{4}$
 $\frac{n}{8}$
 \vdots


not found


 $\frac{n}{4}$
 $\frac{n}{8}$


$\rightarrow 1 \text{ size}$



$$j=0$$

$$e=g$$

$$\text{mid} = \frac{0+9}{2} = 4.5$$

$$j=5$$

$$e=g$$

$$\text{mid} = \frac{1+5}{2} = 3$$

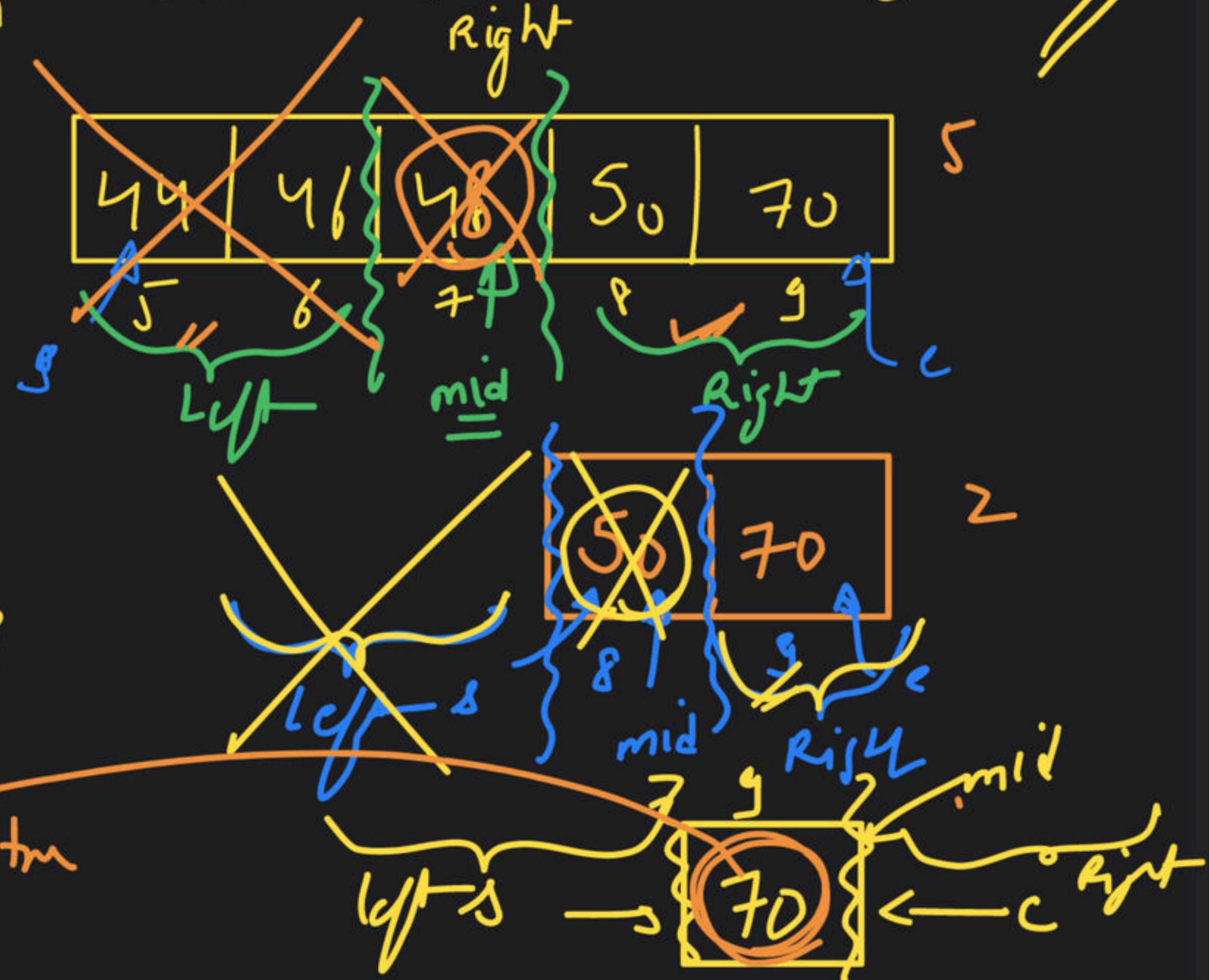
$$j=8, e=g$$

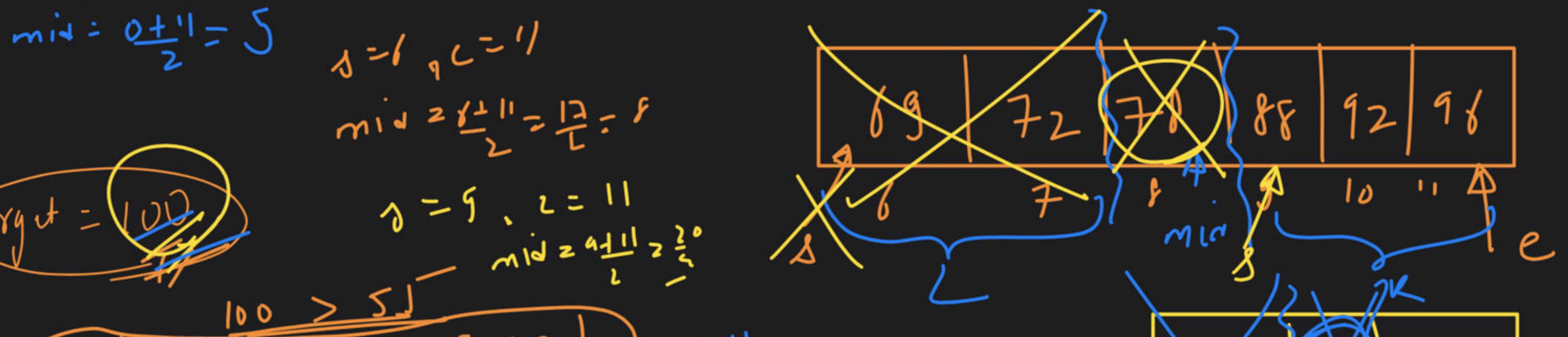
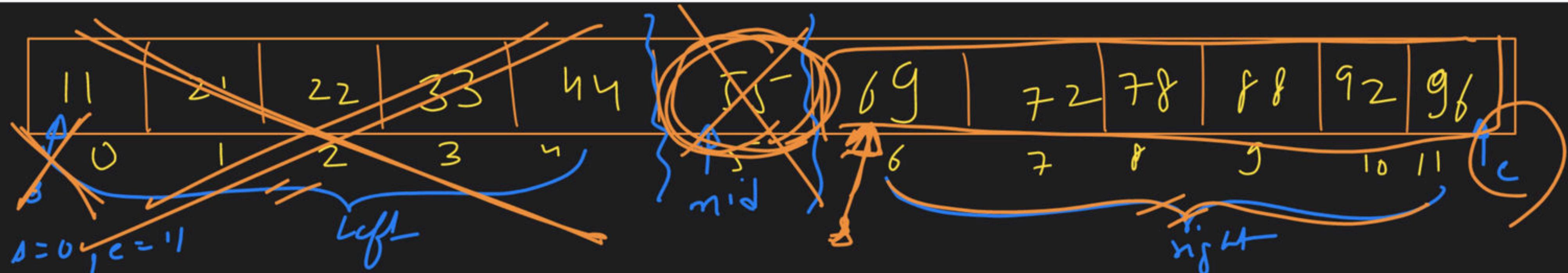
$$\text{mid} = \frac{8+9}{2} = 8.5 = 9$$

$$j=9, e=g$$

$$\text{mid} = \frac{9+9}{2} = 9$$

return tm

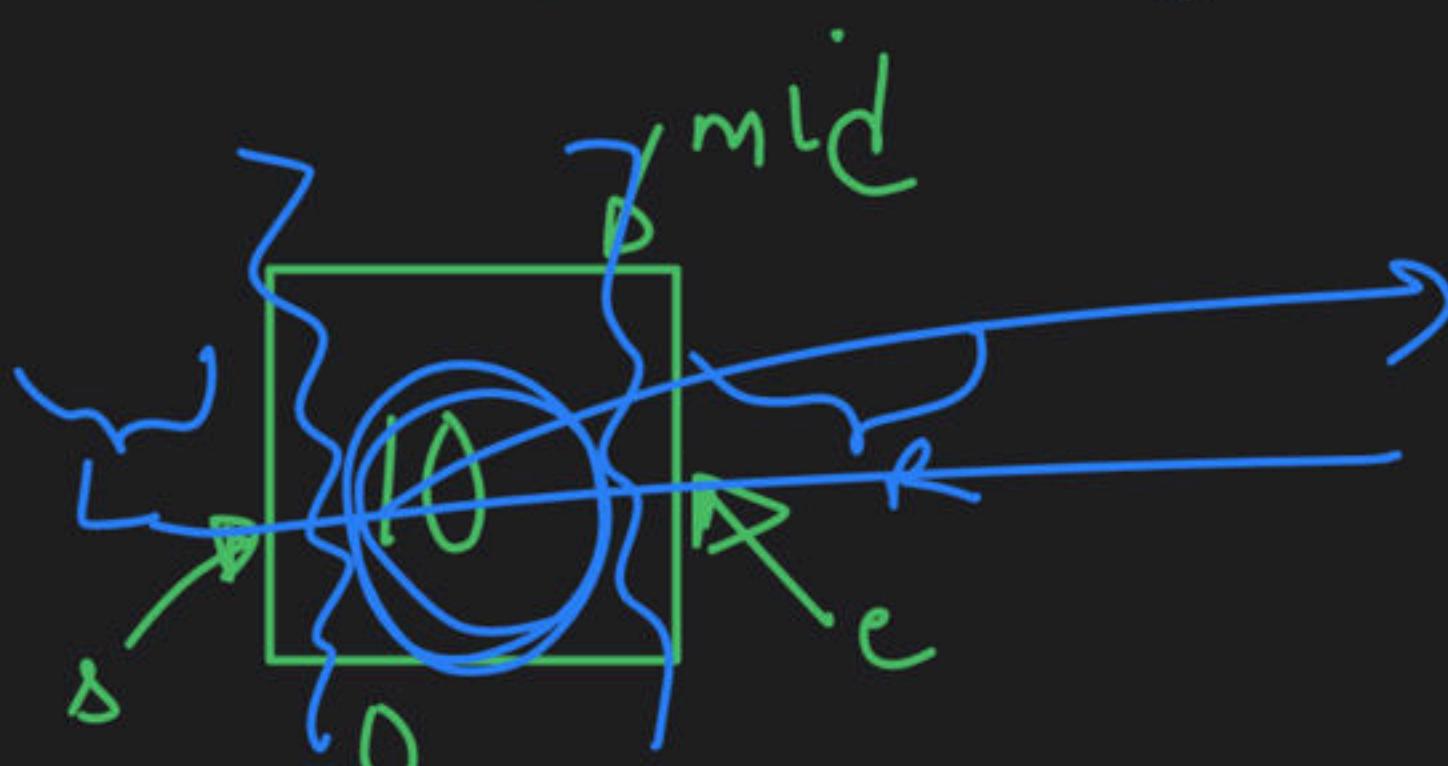
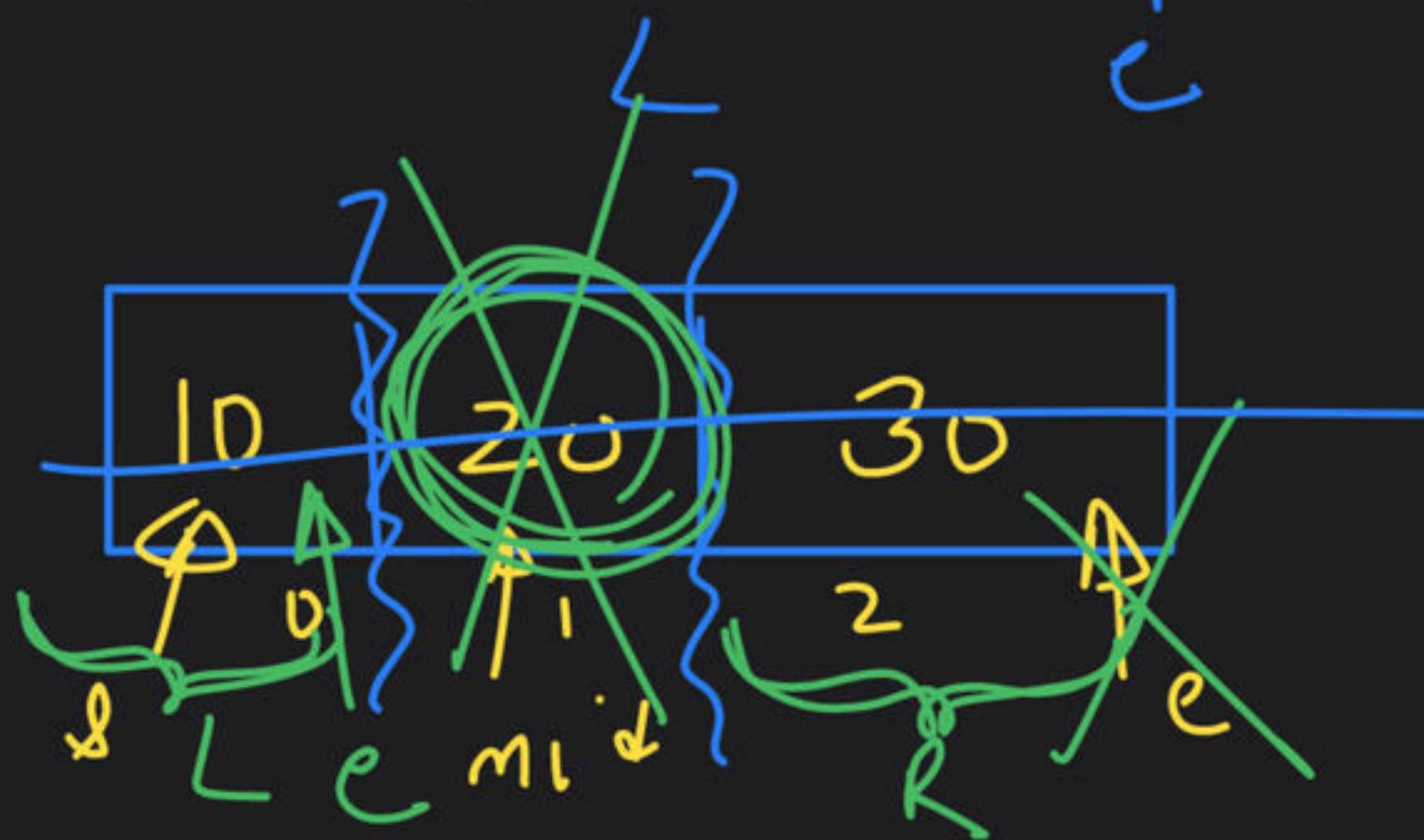
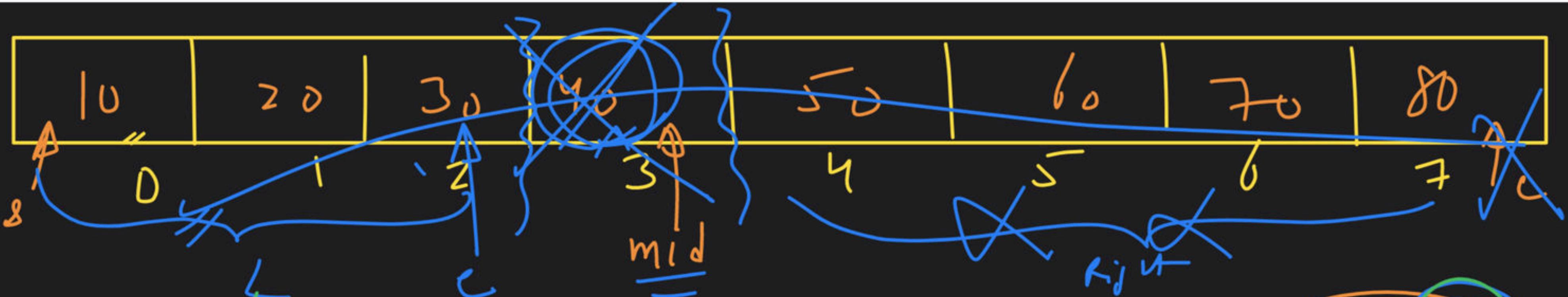




if ($arr[mid] == target$)
return true

if ($target > arr[mid]$)
{
 // target in arr
 b = mid + 1
}

if ($target < arr[mid]$)
{
 // left me arr
 c = mid - 1
}



$$s = 0, e = 2$$

$$\text{mid} = \frac{0+2}{2} = 1$$

$$s = 0, e = 0$$

$$\text{mid} = \frac{0+0}{2} = 0$$



$$s = 0, e = 1$$

$$\text{mid} = \frac{0+1}{2} = 0$$

return true
invalid arr if $s > e$
 $s \leq e \Rightarrow$ tab take chalne hai left me jao
if (target < arr[mid])
 $e = \text{mid} - 1$

$s > e \rightarrow$ invalid array \rightarrow nukta hai

$s \leq e \rightarrow$ valid array \rightarrow chalta hai

if ($arr[mid] == target$) \rightarrow return true;

if ($target > arr[mid]$) \rightarrow right me jao
 $s = mid + 1$

if ($target < arr[mid]$) \rightarrow

right me jao

$e = mid - 1$

assumed

inc bdcv
array

Binary
Search

int start = 0; int end = n - 1;

int mid = $\frac{(start + end)}{2}$;

while (start <= end)

{

if (arr[mid] == target)

return true;

if (target < arr[mid])

e = mid - 1;
;

if (target > arr[mid])

s = mid + 1;
;

mid = $\frac{(start + end)}{2}$

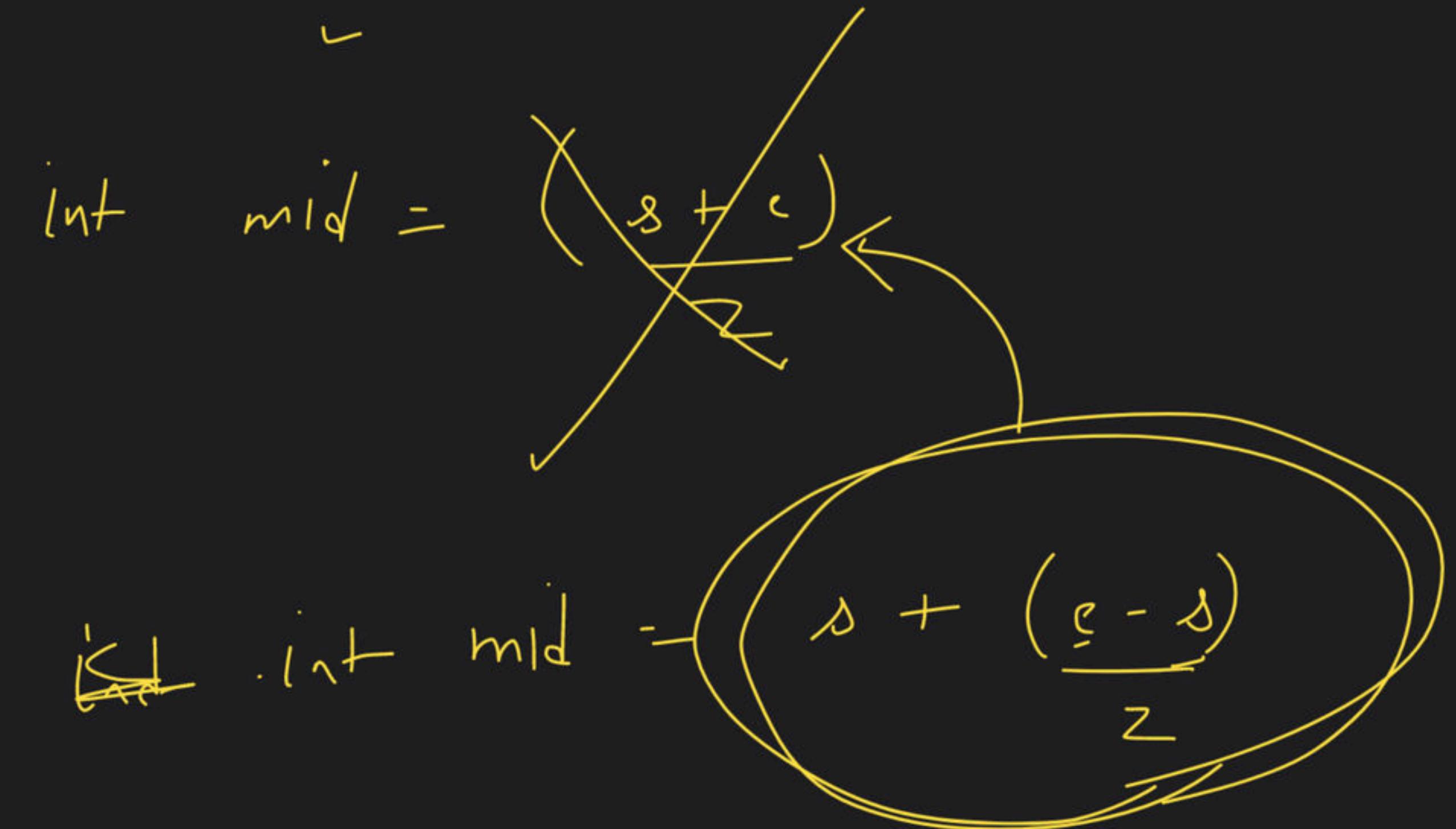
}

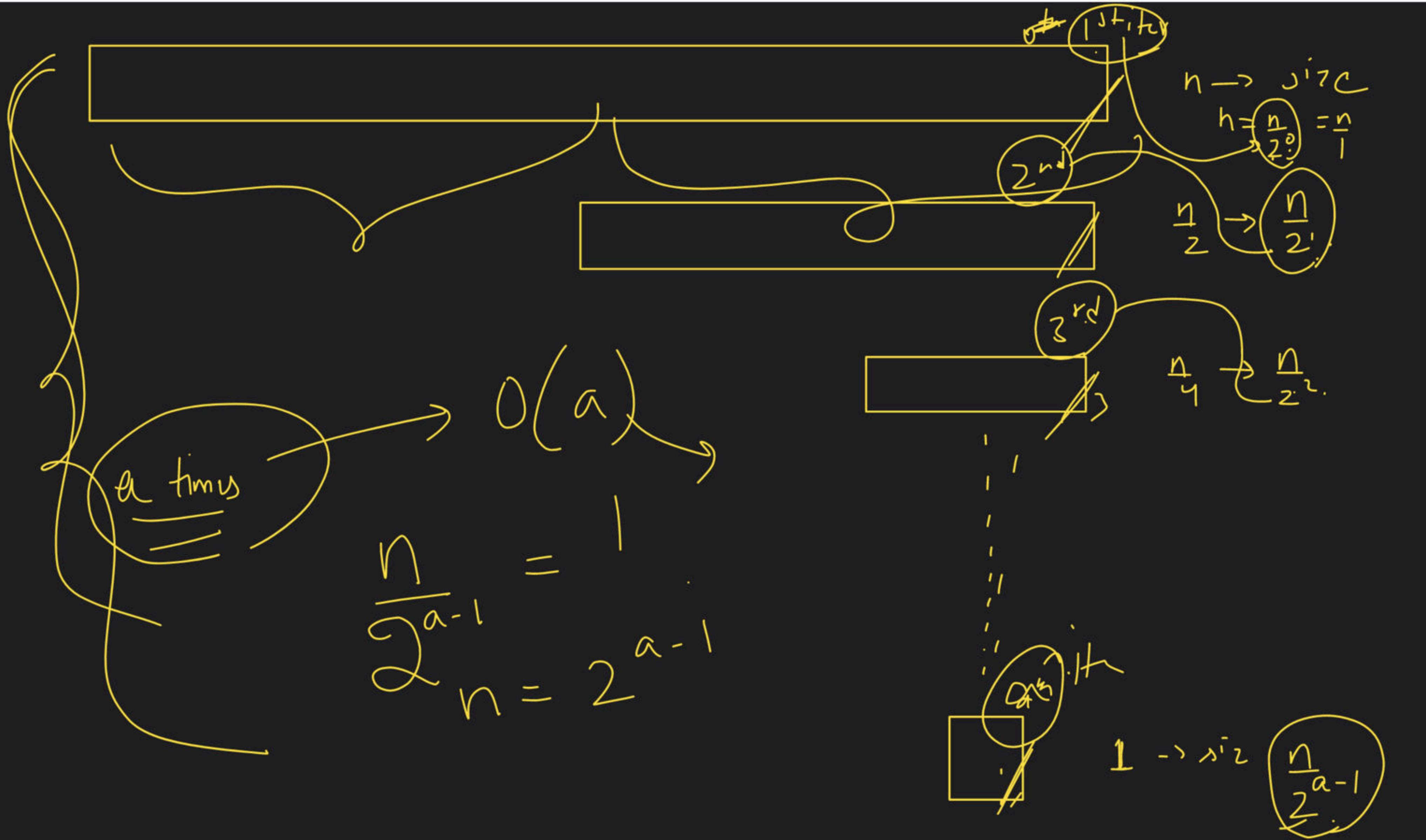


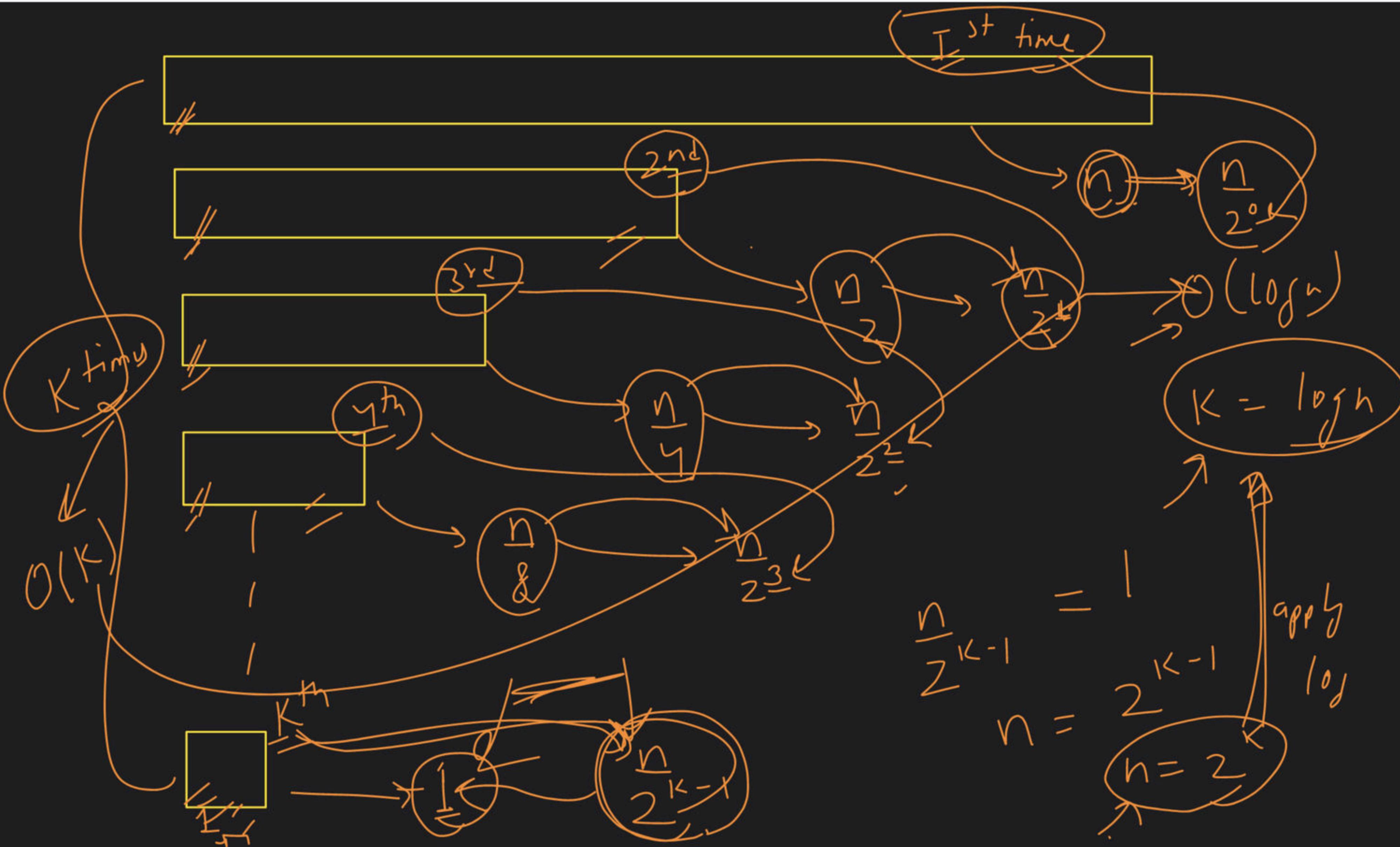
$$mid = s + \frac{e-s}{2};$$

$$int mid = \frac{s+e}{2};$$

Acche
Bacche





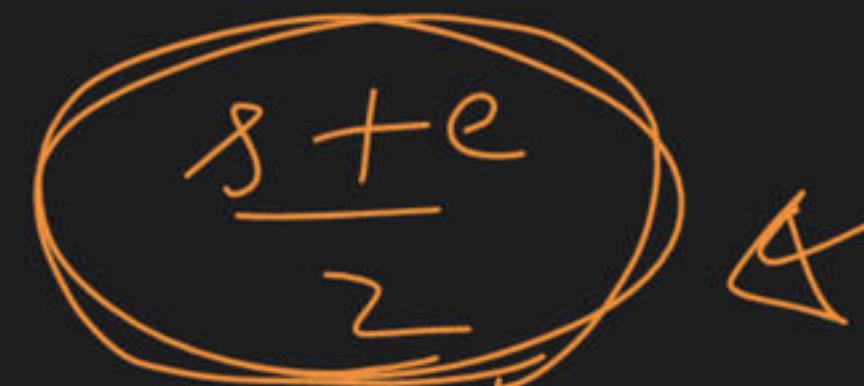




$$\beta^+ - \frac{e}{2} - \frac{\gamma}{2}$$

$$\beta^+ - \frac{\Delta}{2} + \frac{e}{2}$$

$$\frac{\beta}{2} + \frac{e}{2}$$



$$n = 2^{a^{-1}}$$

$$\rightarrow n = \underline{2^a}$$

$$n = 2^{a^{-1}}$$

app^b log

$$\log_2 n = \log_2(2^a)$$

$$\log n = a$$

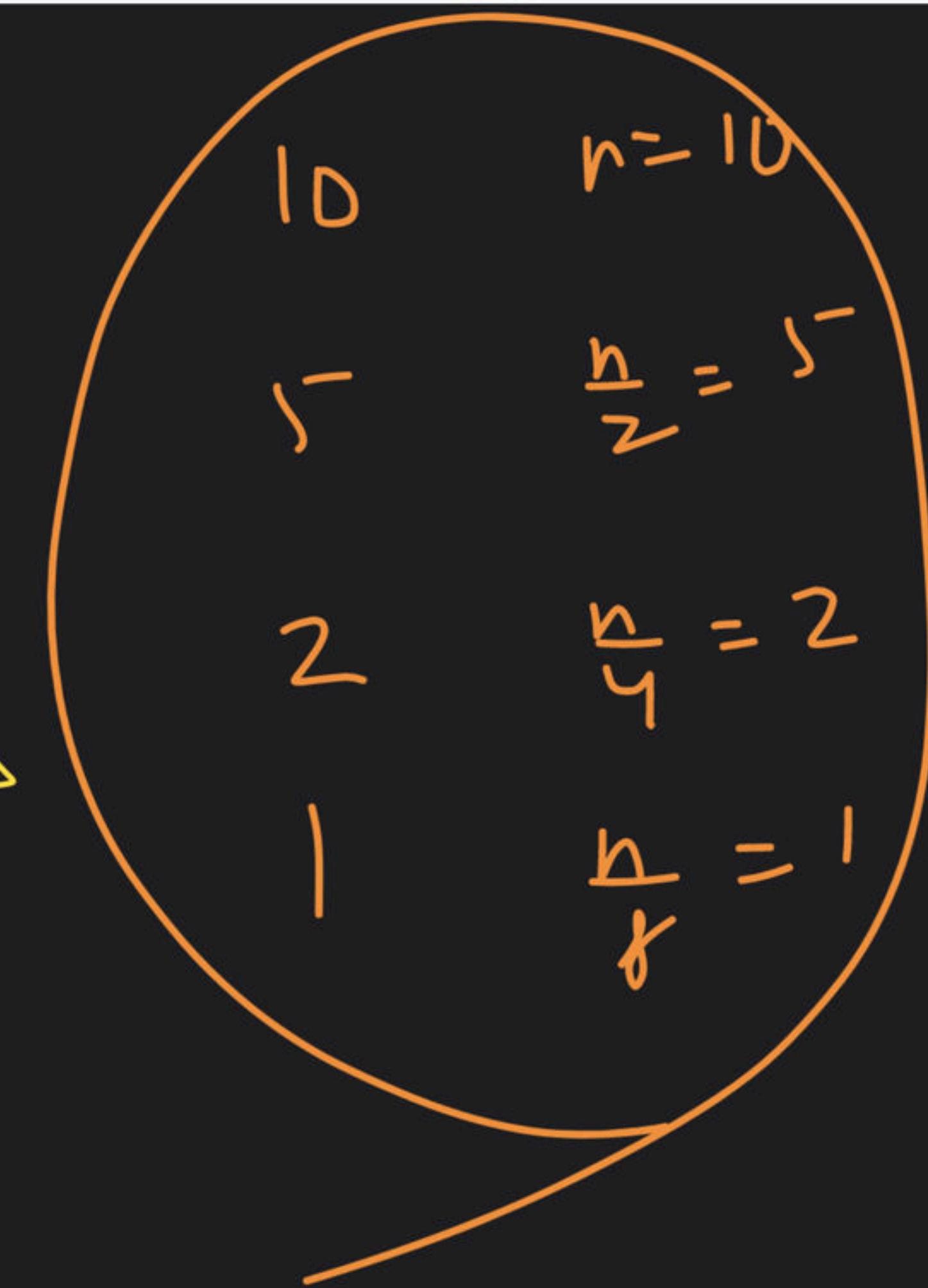
$$O(a)$$

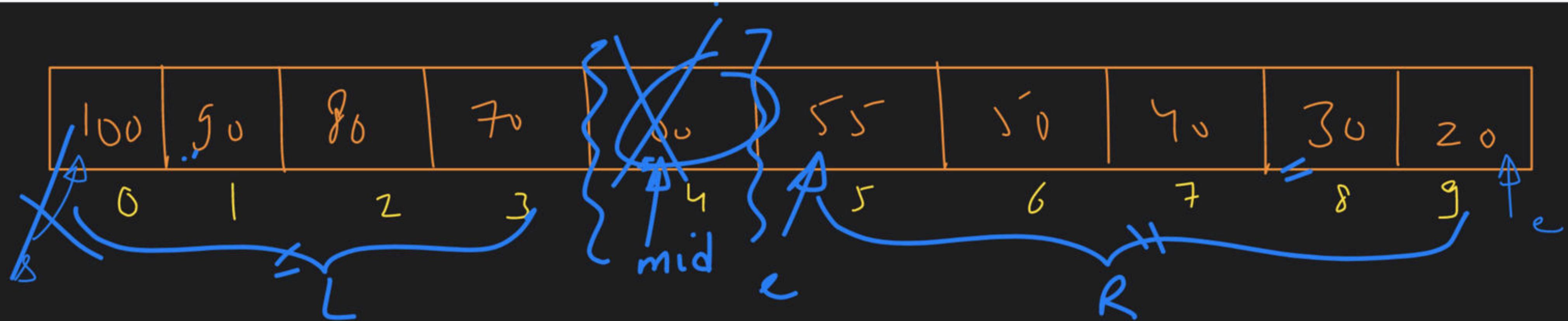
$$O(\log n)$$

$$\Gamma \cdot C$$



invalid array \rightarrow $i > c$





target
90

$l = 0$
 $e = 9$

$$mid = \frac{9}{2} = 4$$

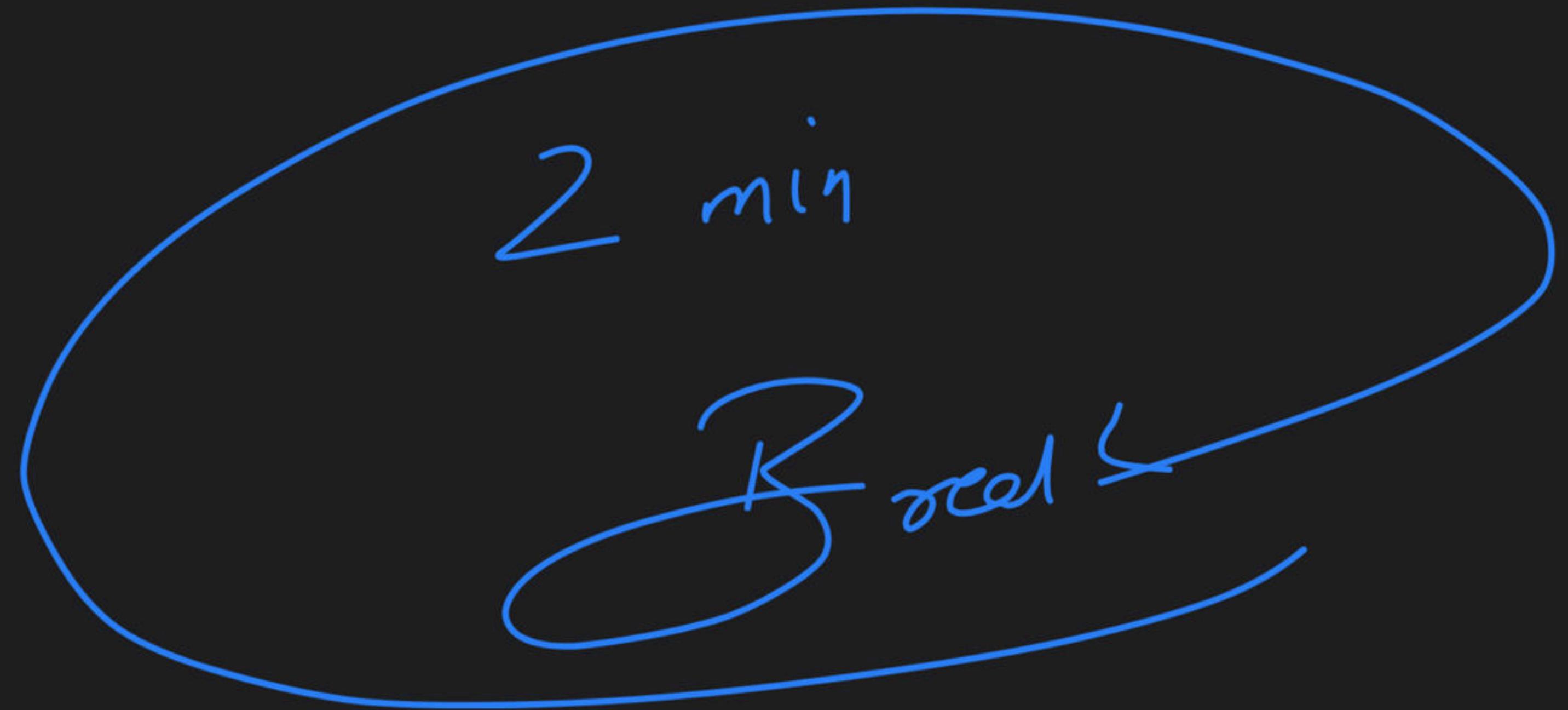
increasing
order

dec order

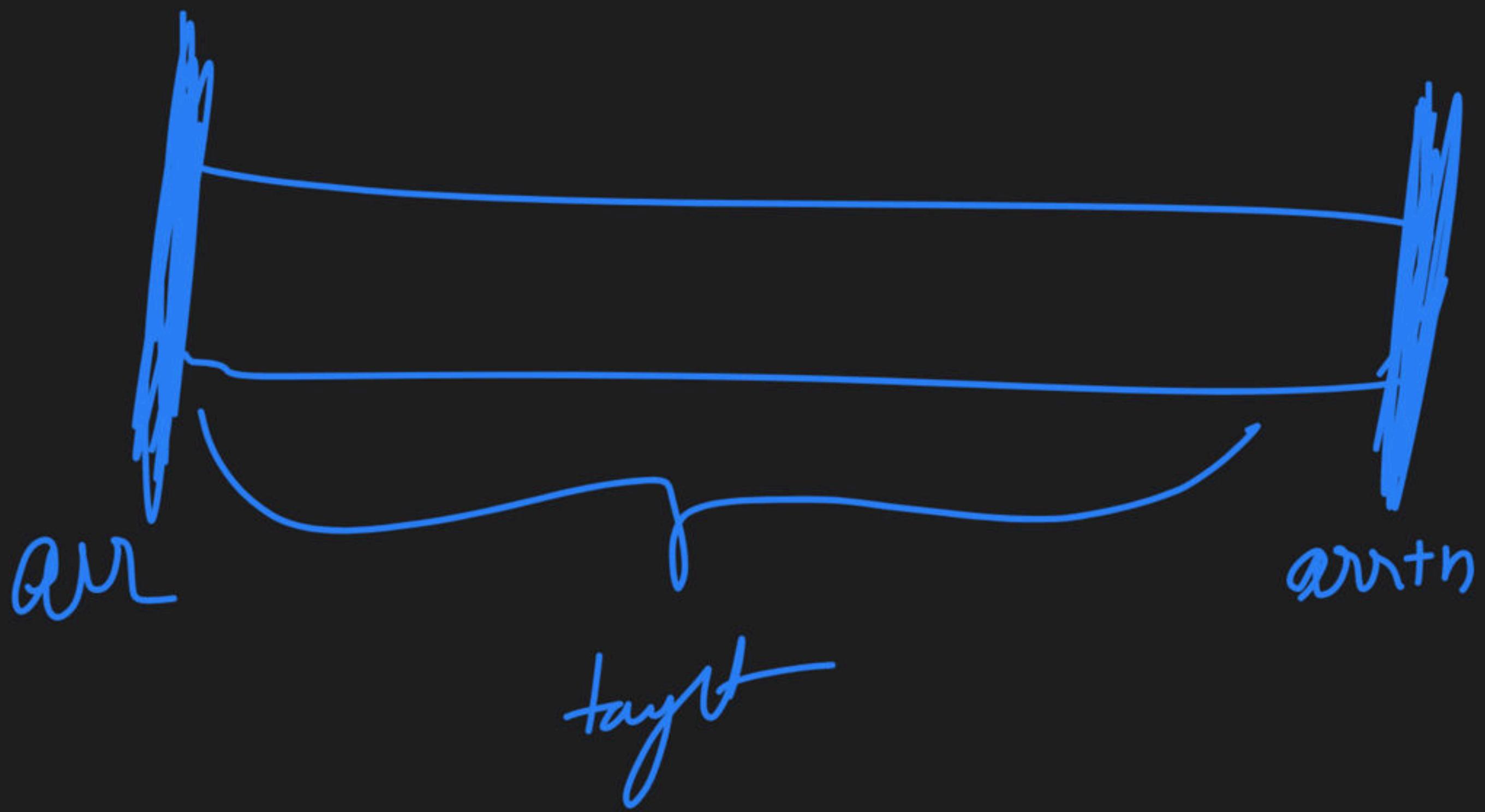
if ($target < arr[mid]$)
right = mid

$$l = mid + 1$$

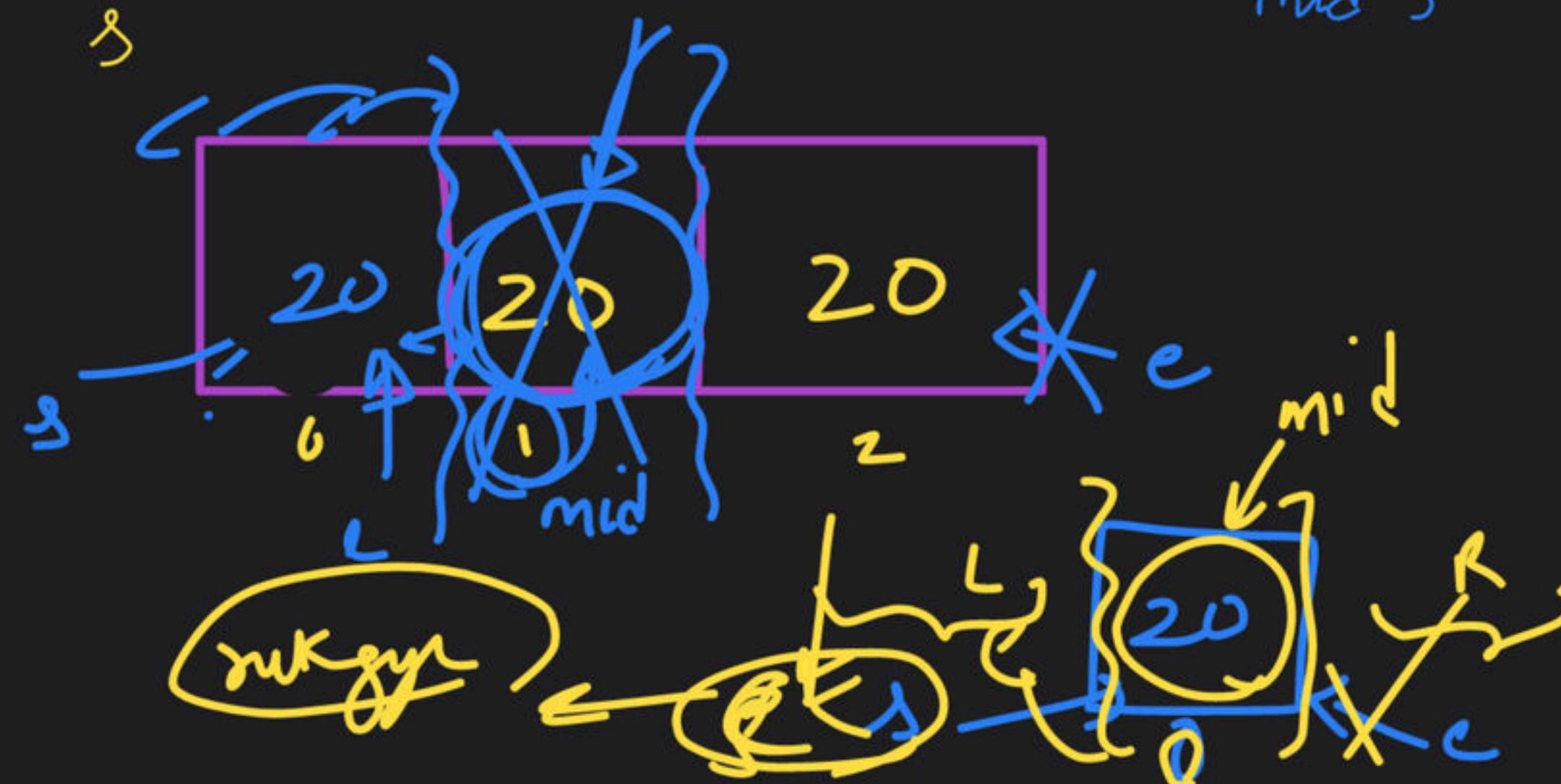
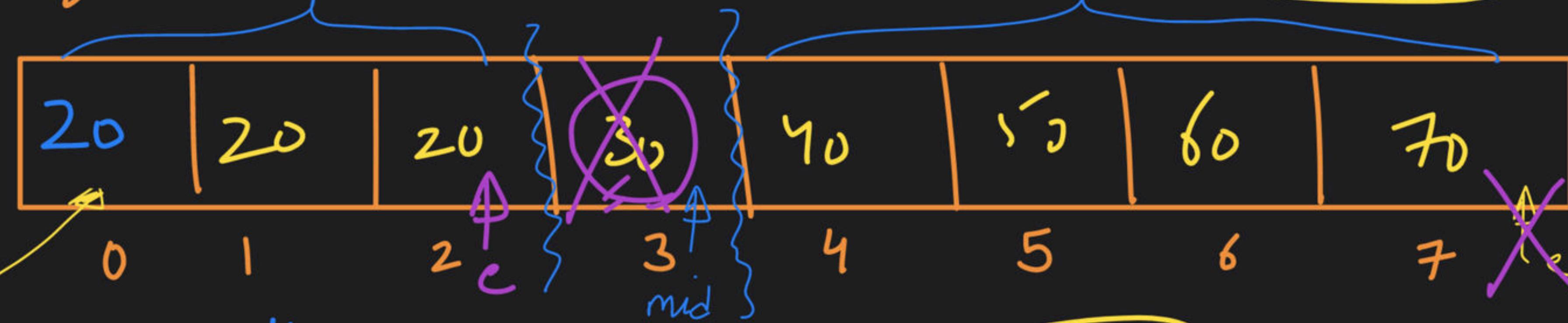
dec order
if ($target > arr[mid]$)
left = mid - 1



SL



→ find first
~~f =~~
Occurrence → (index return)
 ansIndex = ~~X0~~
 target = 20



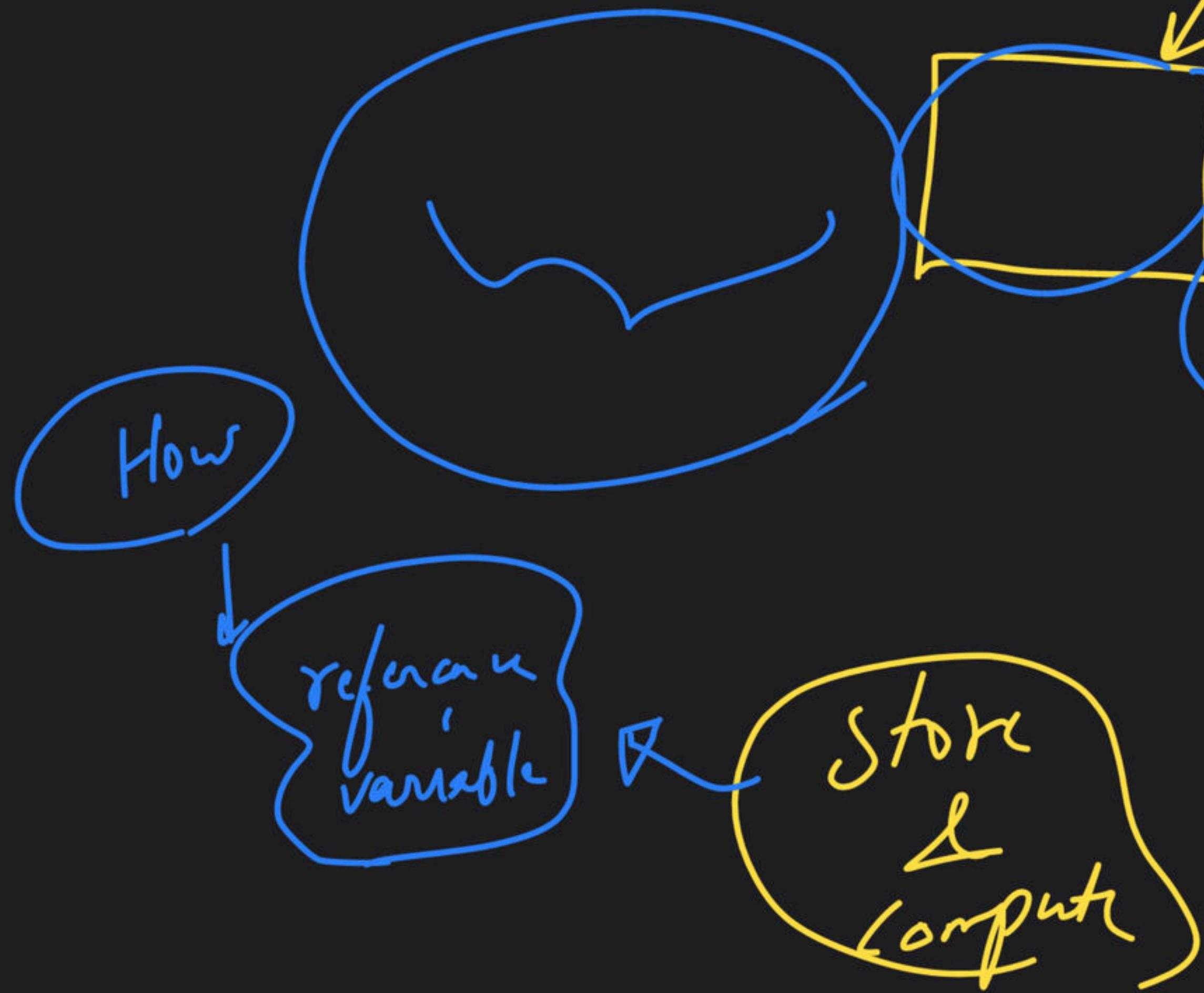
ans found
 first Occ or
 not first Occ
 store & compute

aH store

ans dhund lia hai

final As ho
bhi okh
h

or
final As nahi bli
ho skta
tar



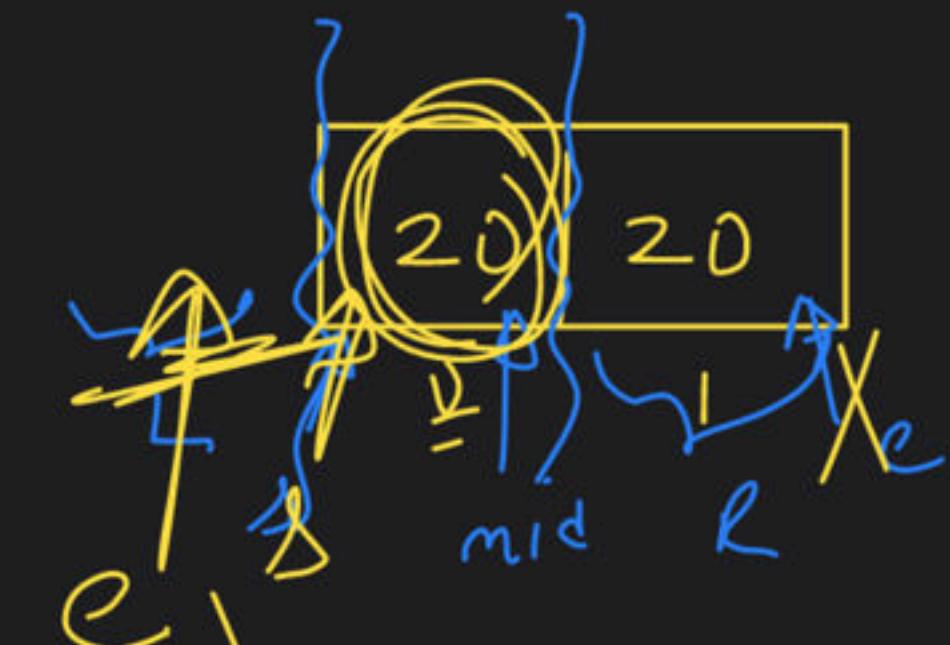
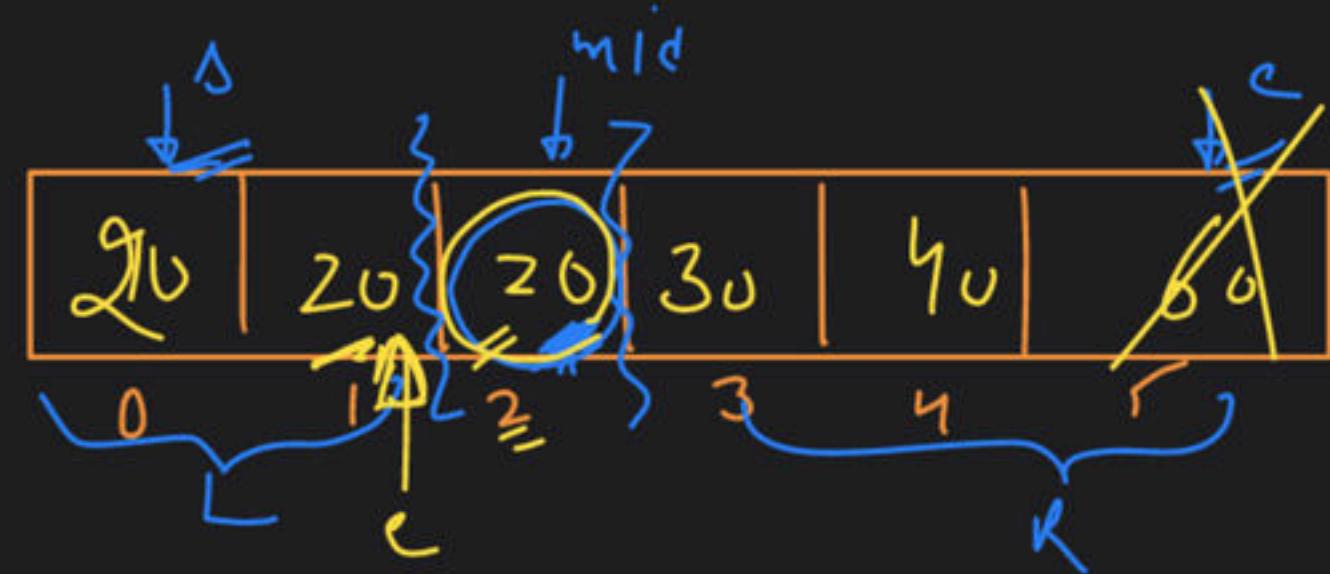
```

void findFirstOcc (arr, n, target, ansIndex)
{
    int s = 0, e = n - 1; int mid = s + (e - s) / 2;
    while (s <= e)
    {
        if (target == arr[mid])
        {
            ansIndex = mid;
            e = mid - 1;
        }
        else if (target > arr[mid])
            s = mid + 1;
        else
            e = mid - 1;
        mid = s + (e - s) / 2;
    }
}

```

\downarrow
 may or may not be
 first occurrence

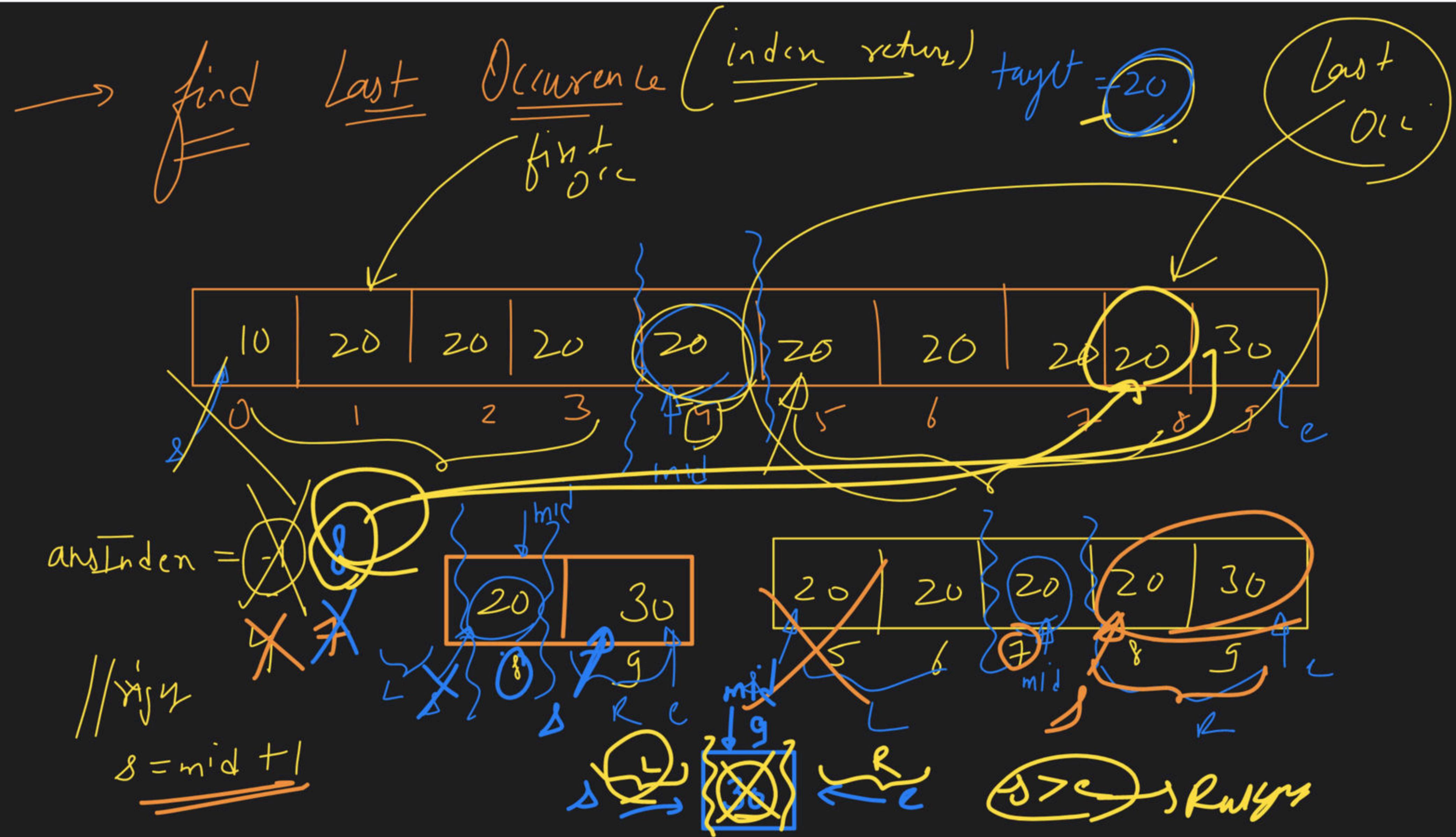
\downarrow
 store
 compute



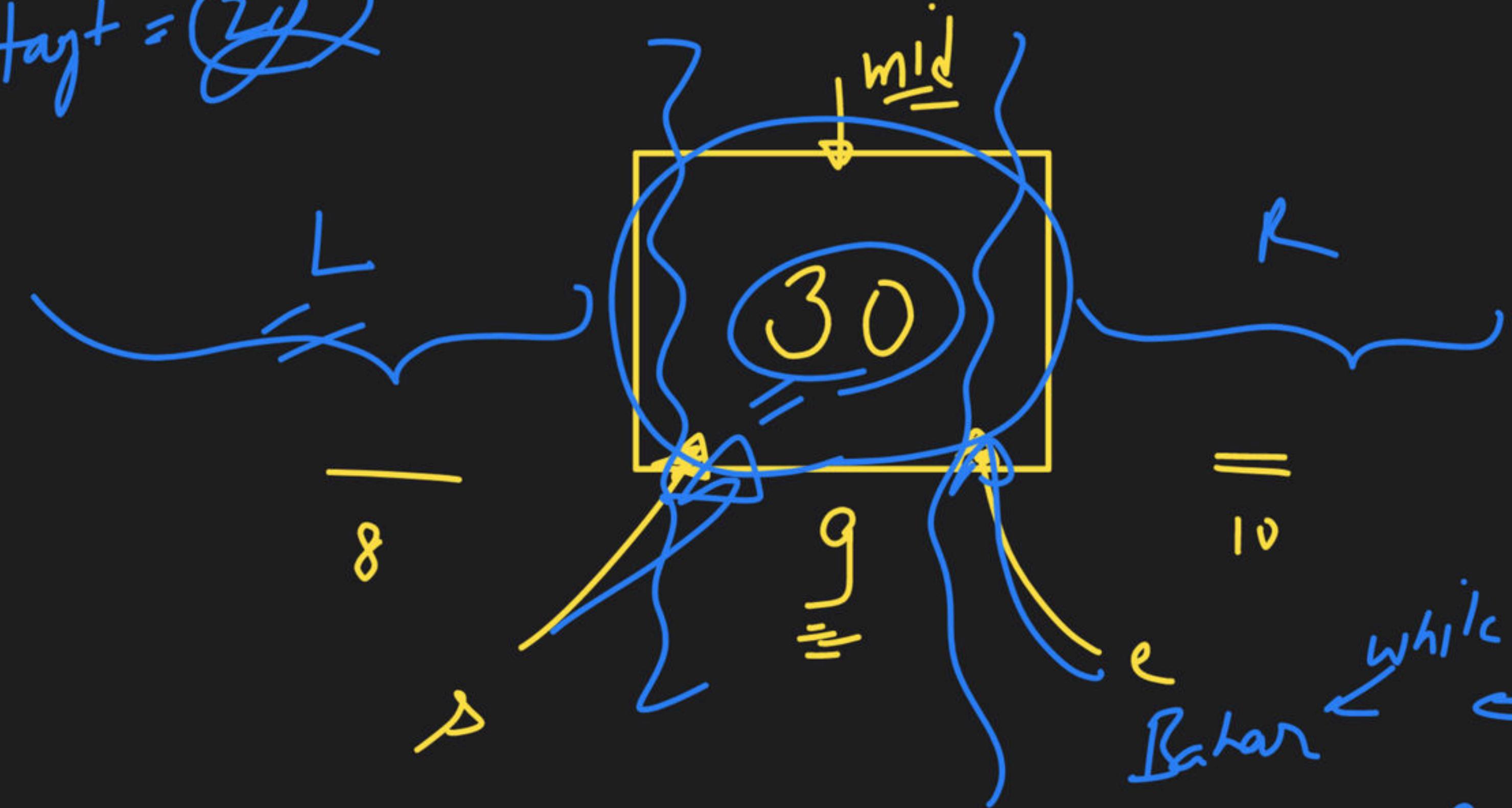
\downarrow
 $s > e$

\downarrow
 stuck

3



~~tag⁺ = 22~~



$$l = 9$$
$$c = 9$$

$$\text{mid} = \frac{9+9}{2}$$

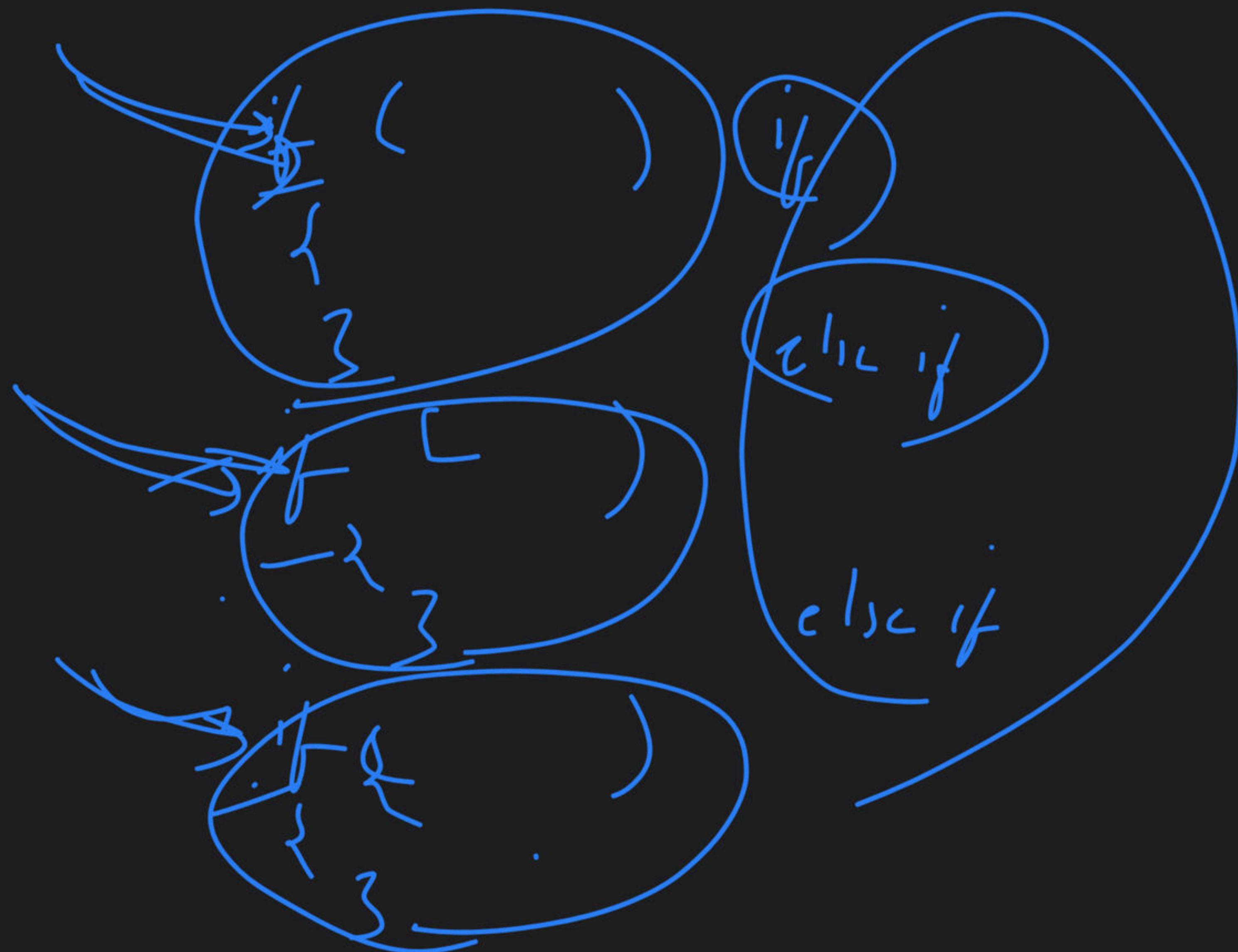
$$= 9$$

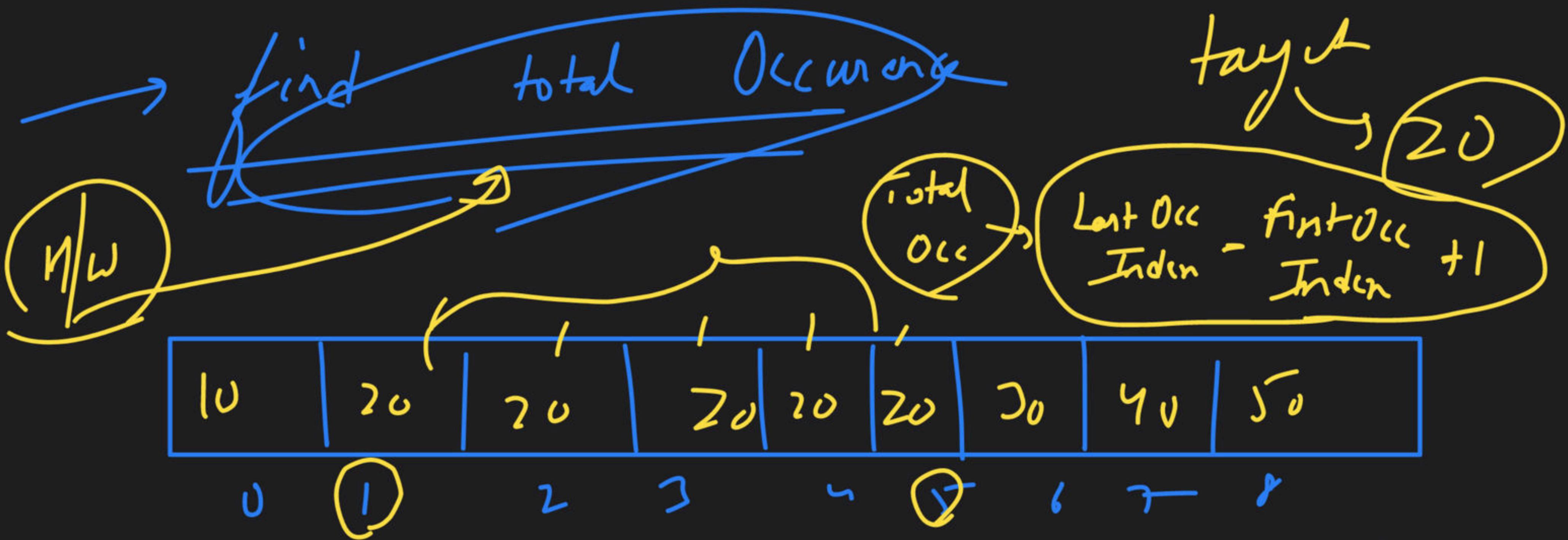
left max

$$c = \text{mid} - 1 \Rightarrow c = 9 - 1$$

$$c = 8$$



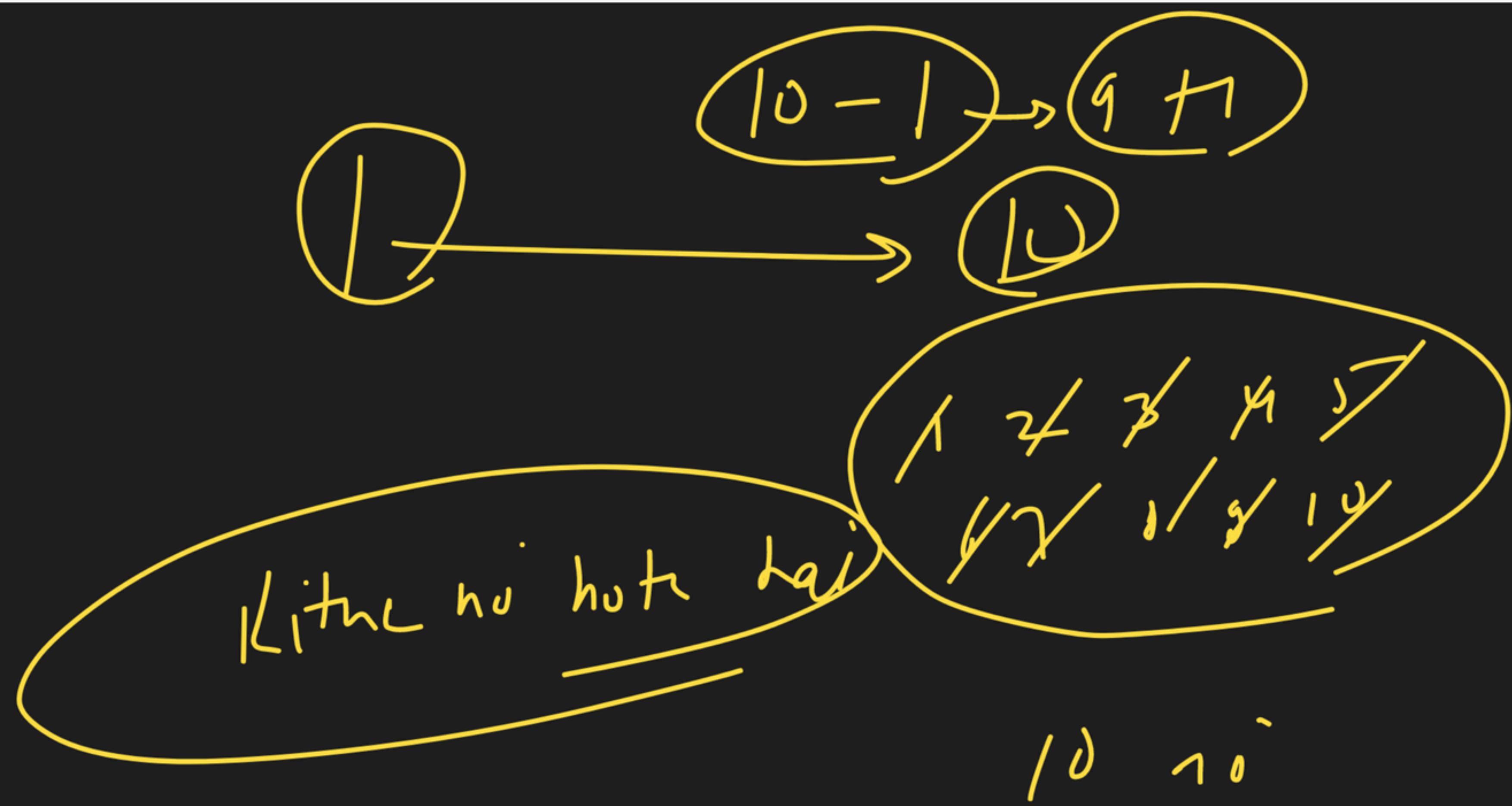




$O(\log n) \rightarrow \text{first Occ} \rightarrow \text{index} \rightarrow 1$
 $O(\log n) \rightarrow \text{last Occ} \rightarrow \text{index} \rightarrow 5$

$$\begin{aligned}
 & \text{Total Occ} \\
 & (5 - 1) + 1 \\
 & 4 + 1 = 5
 \end{aligned}$$





$$15 - 11 = 4 + 1$$



~~11~~ ~~12~~ ~~13~~ ~~14~~ ~~15~~

first
j <<

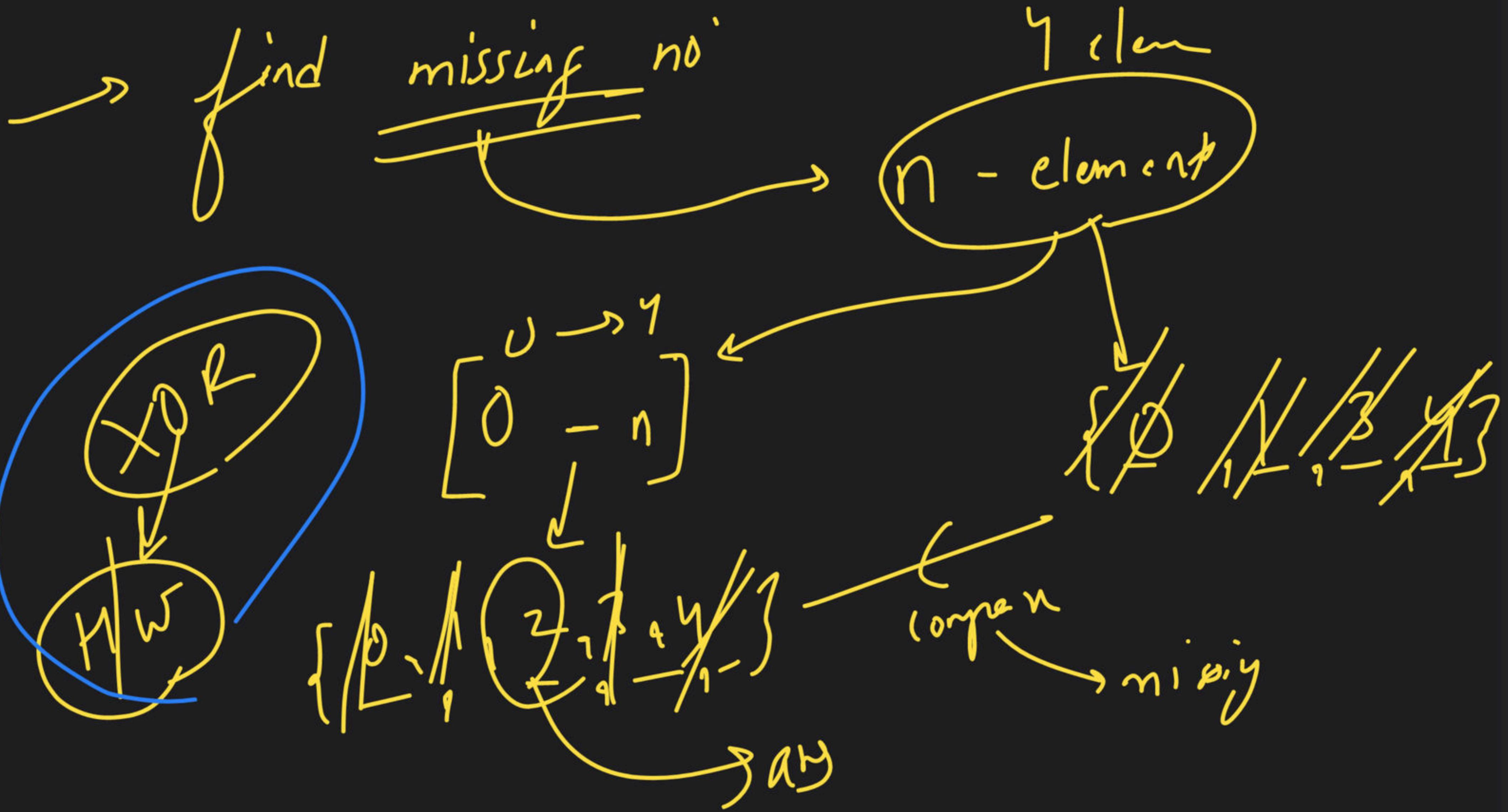


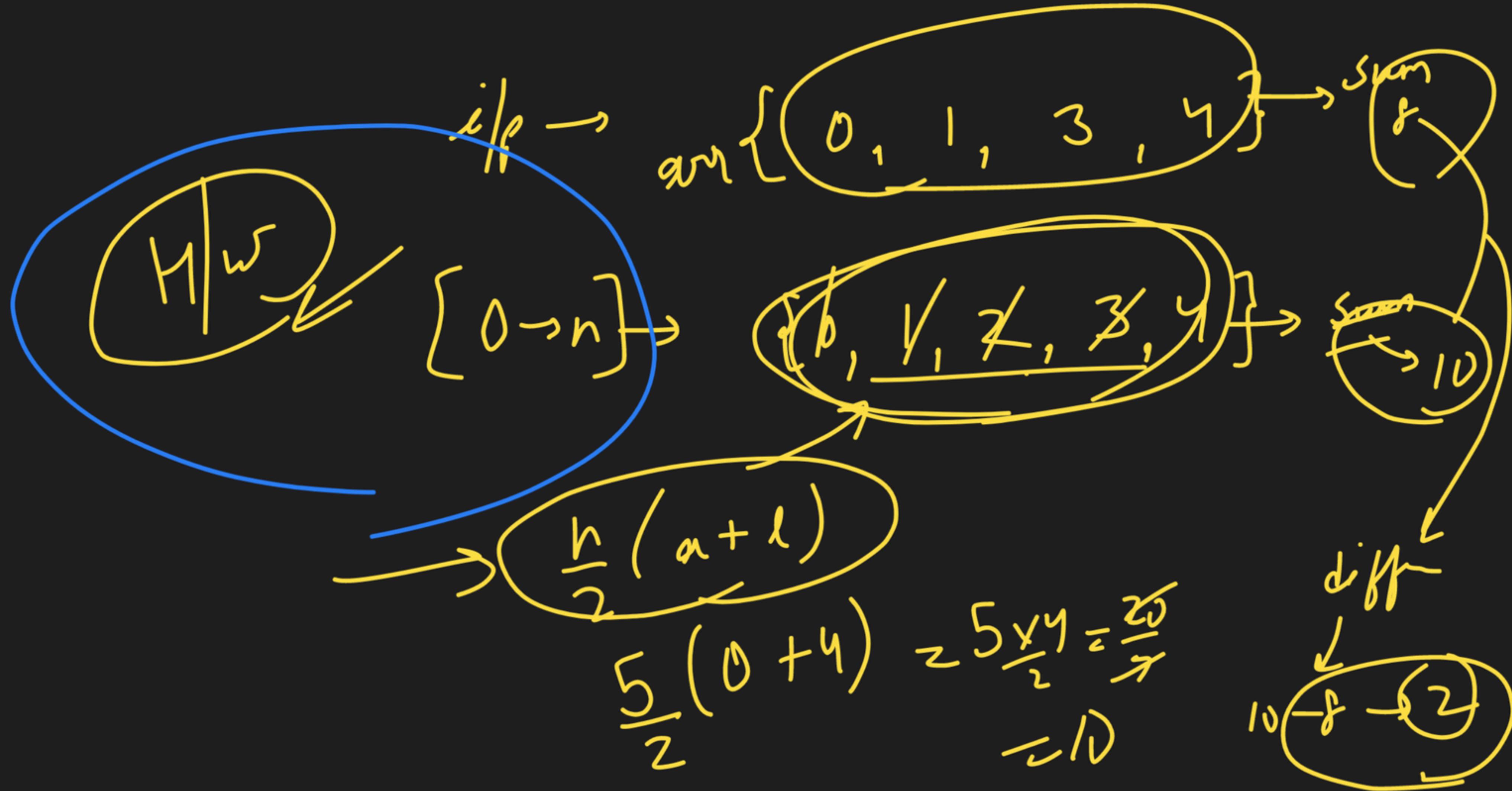
Last() <



Last - first + 1

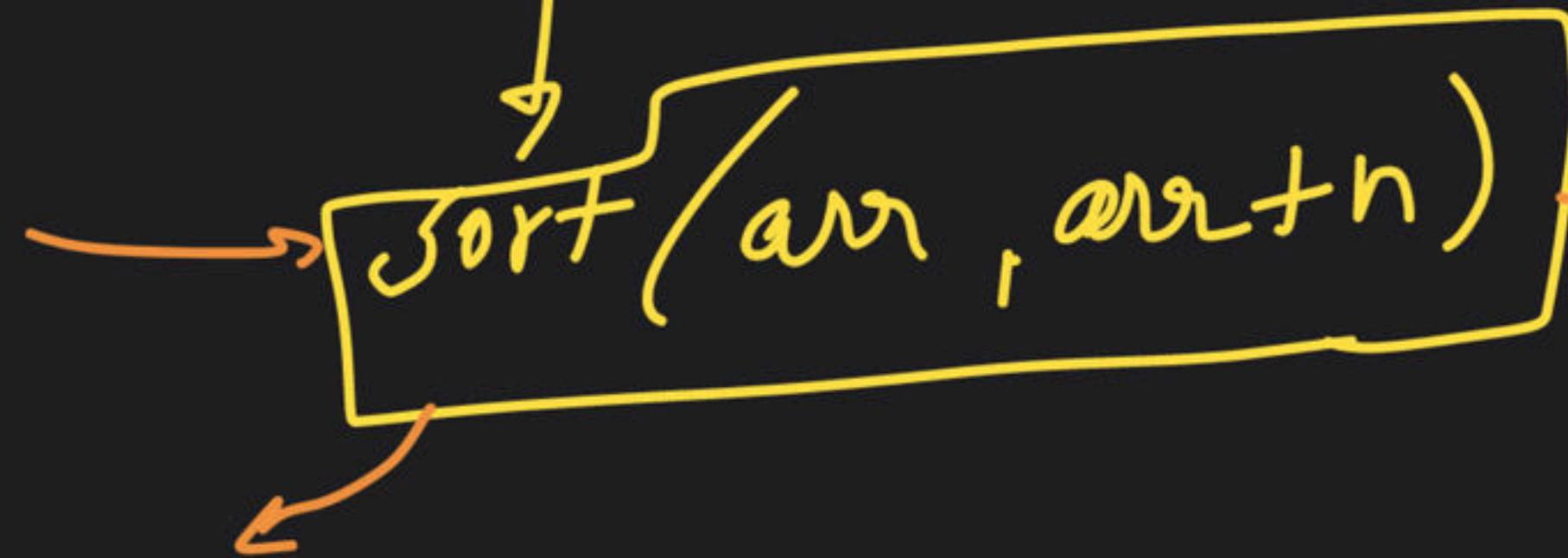








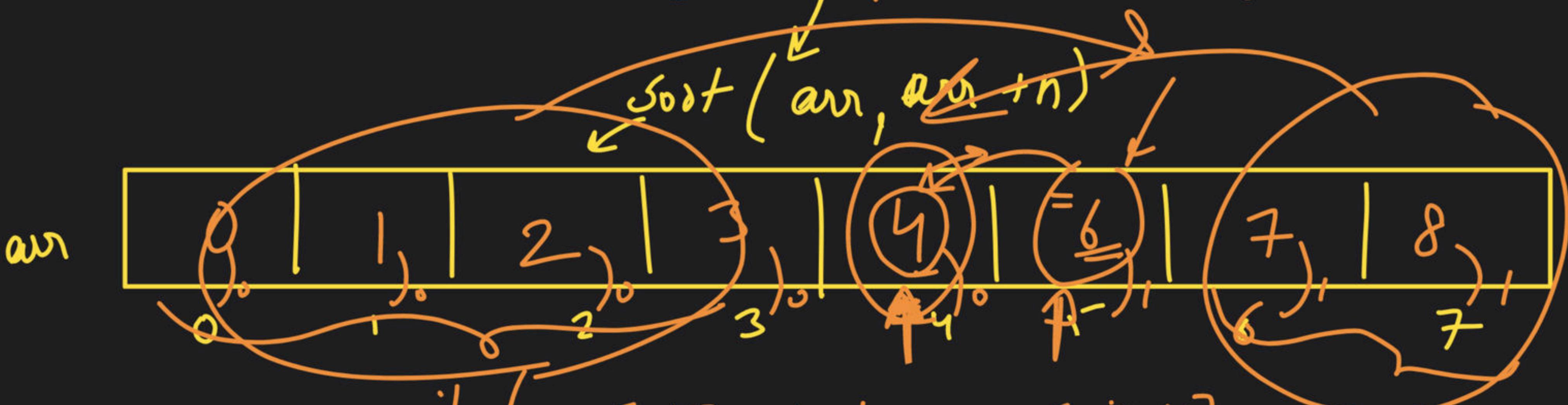
{ 3 , 0 , 1 }



$O(n \log n)$



<i>arr</i>	3	4	6	0	7	2	1	1	8
	0	1	2	3	4	5	6	7	8



if ($\text{arr}[\text{mid}] + 1 \neq \text{arr}[\text{mid} + 1]$)

if ($\text{arr}[\text{mid}] - 1 \neq \text{arr}[\text{mid} - 1]$)

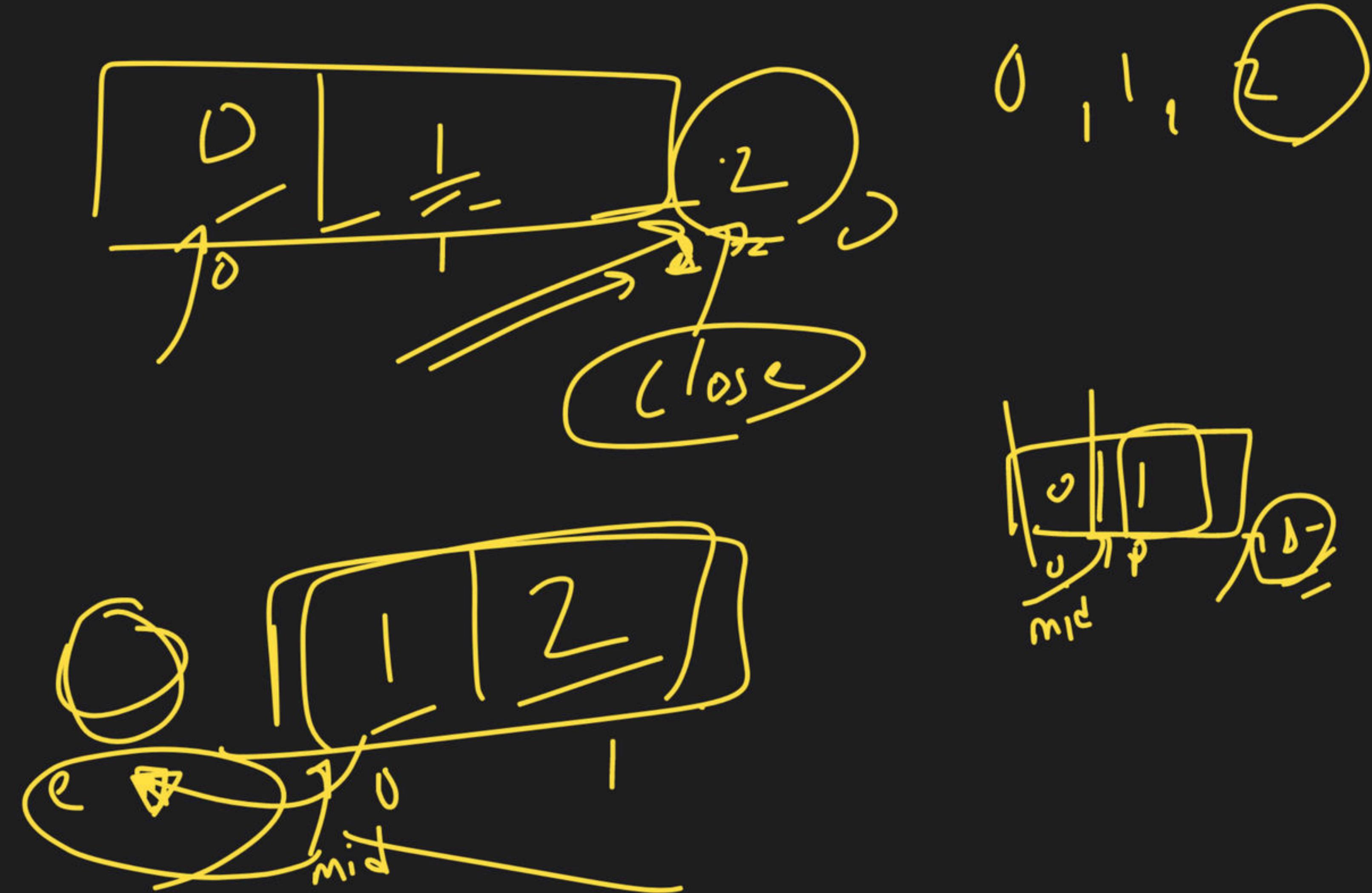
$\text{arr}[\text{mid}] - 1$

$\text{arr}[\text{mid}] + 1 \rightarrow \text{missij} \rightarrow \text{no}$



if ($\text{arr}[\text{mid}+1] - \text{arr}[\text{mid}] \neq 1$)
 ~~$\text{aux}[\text{mid}] + 1$~~

if ($\text{arr}[\text{mid}] - \text{arr}[\text{mid}-1] \neq 1$)
~~return~~
 $\text{arr}[\text{mid}] - 1$



$diff = num - index;$

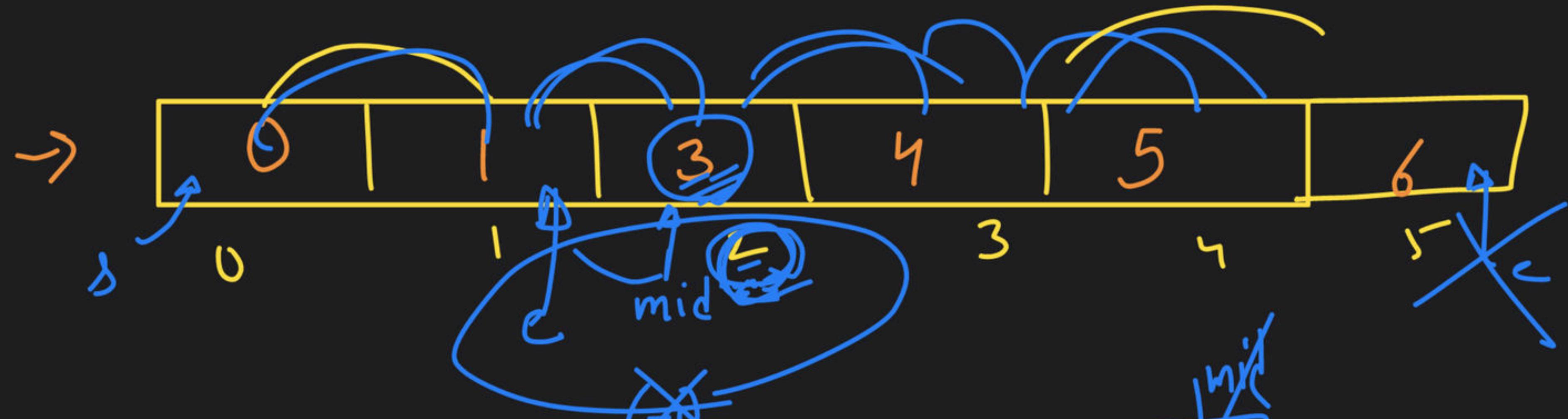
if ($diff == 0$)

 // here $\rightarrow s = mid + 1$

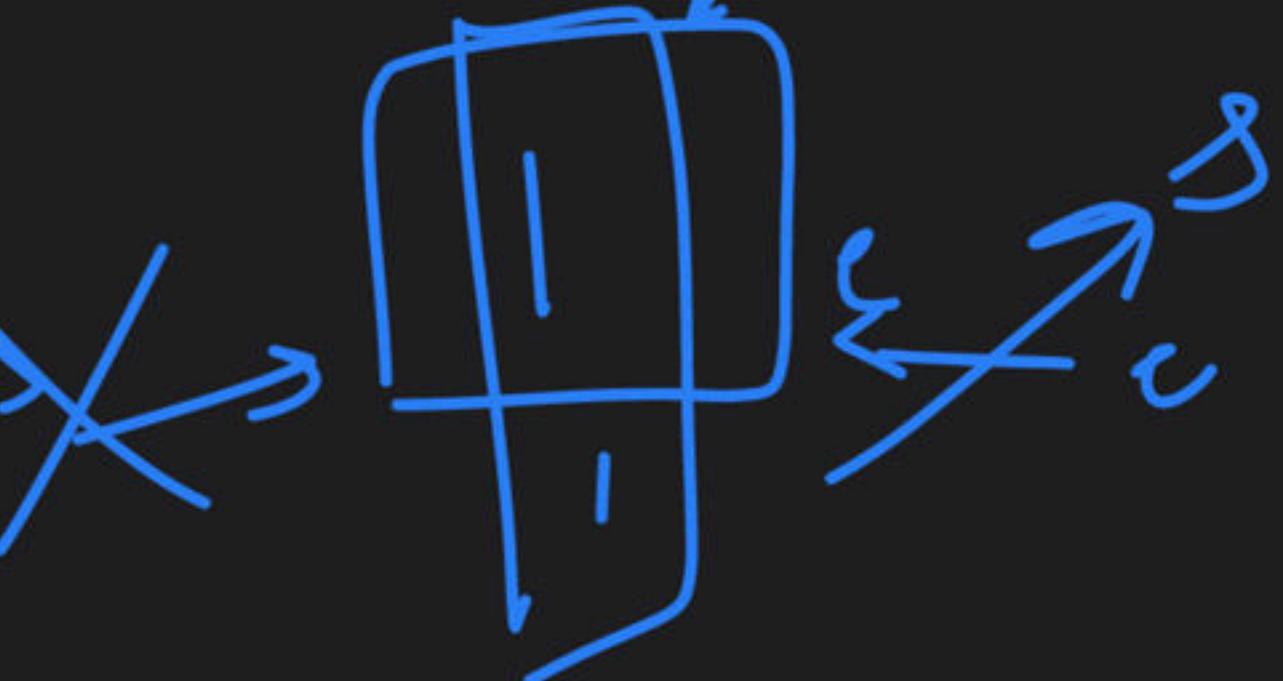
if ($diff == 1$)

 ans = index

 c = mid - 1

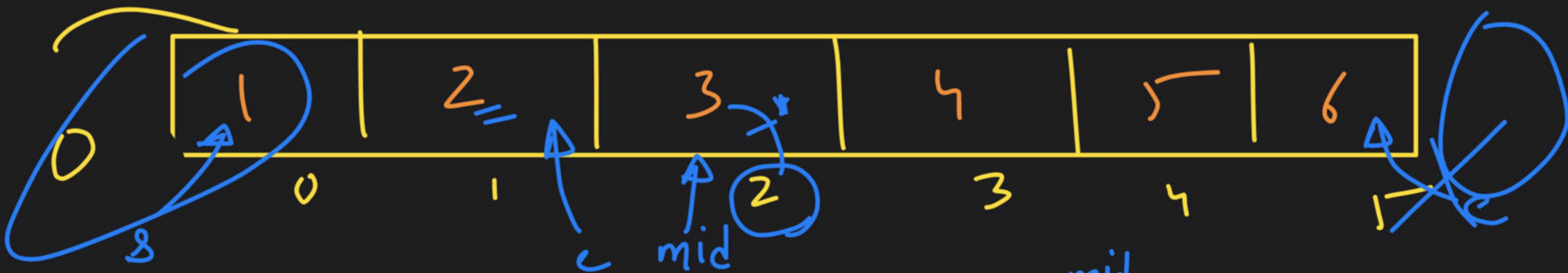


0



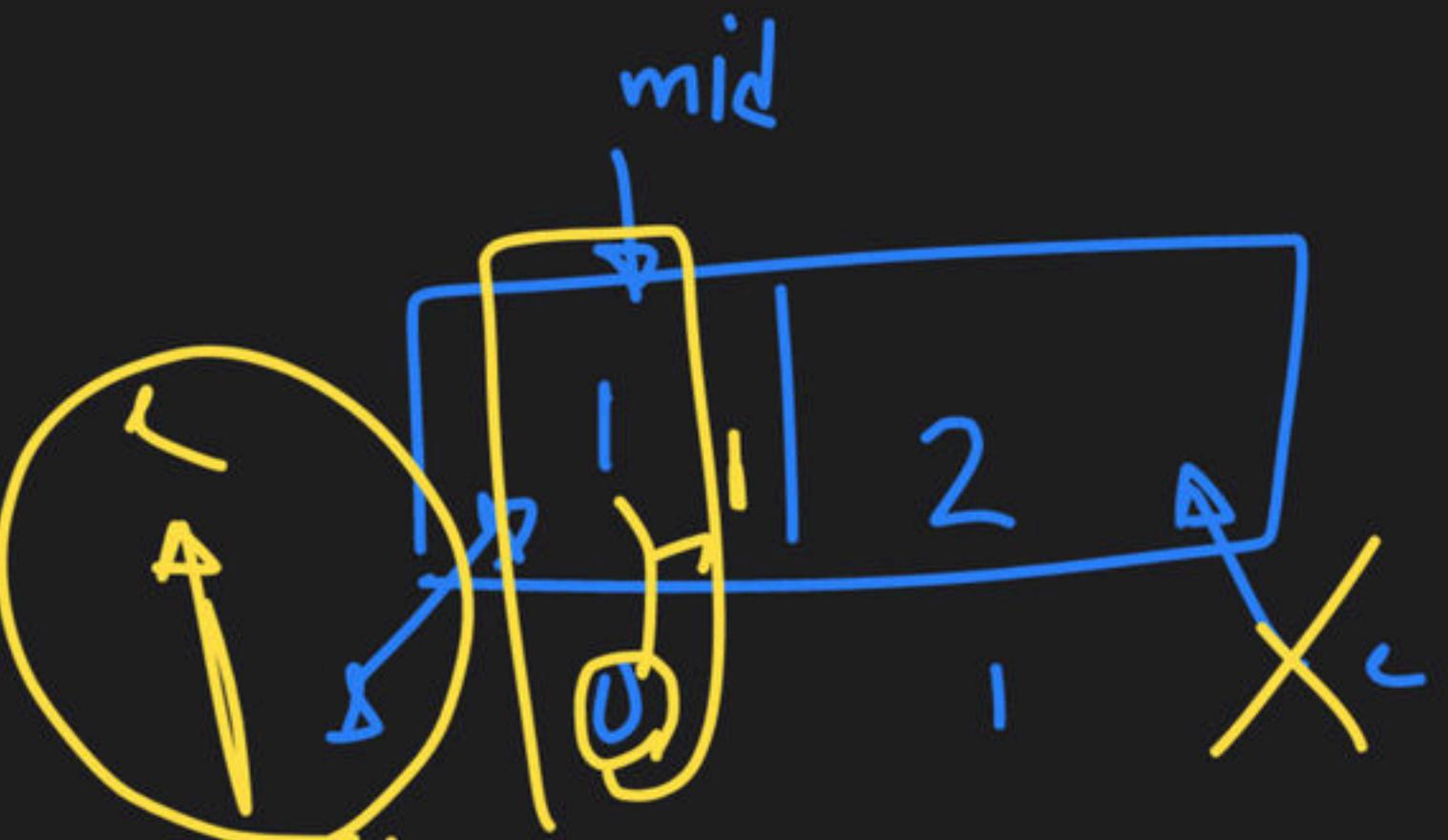
$s > e \Rightarrow$ miss lat

.



~~ans =~~

0



$s > e$ *when tail*



6 mid

