

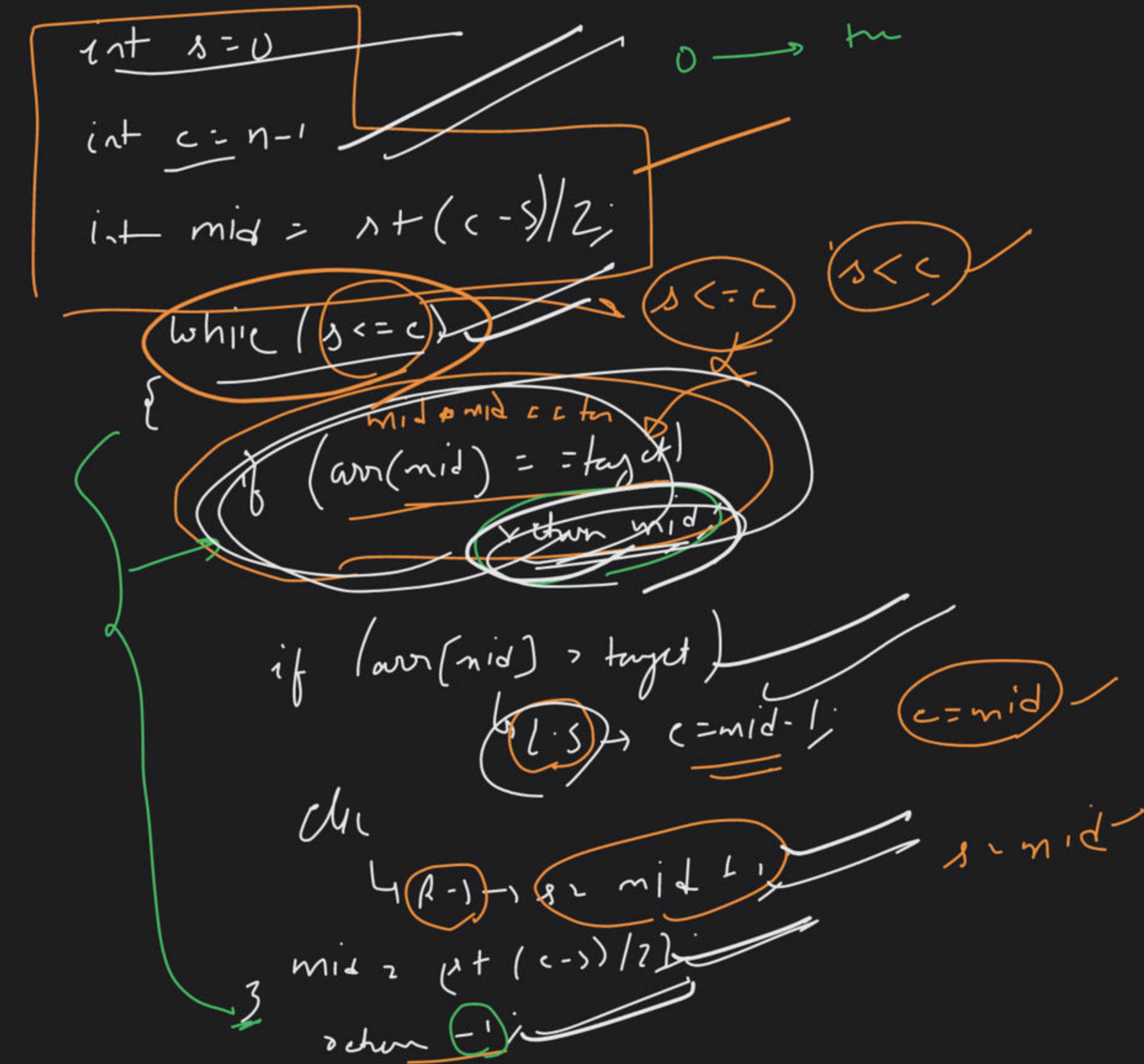
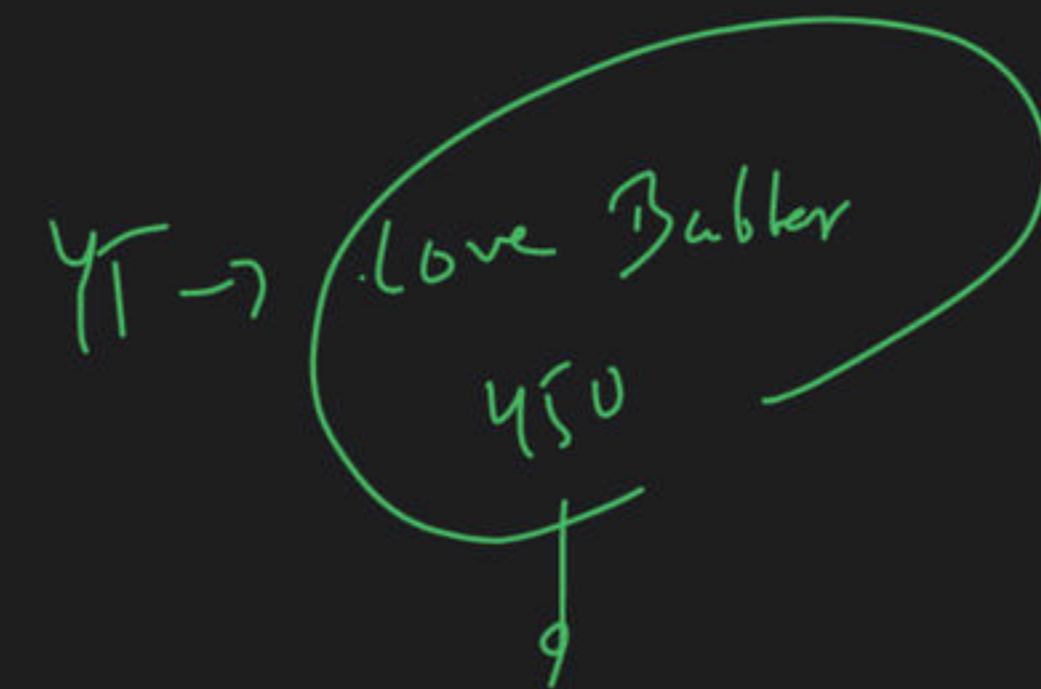
B · S

→ ~~(B · S) | L · B | V · B | T · O | Peak~~



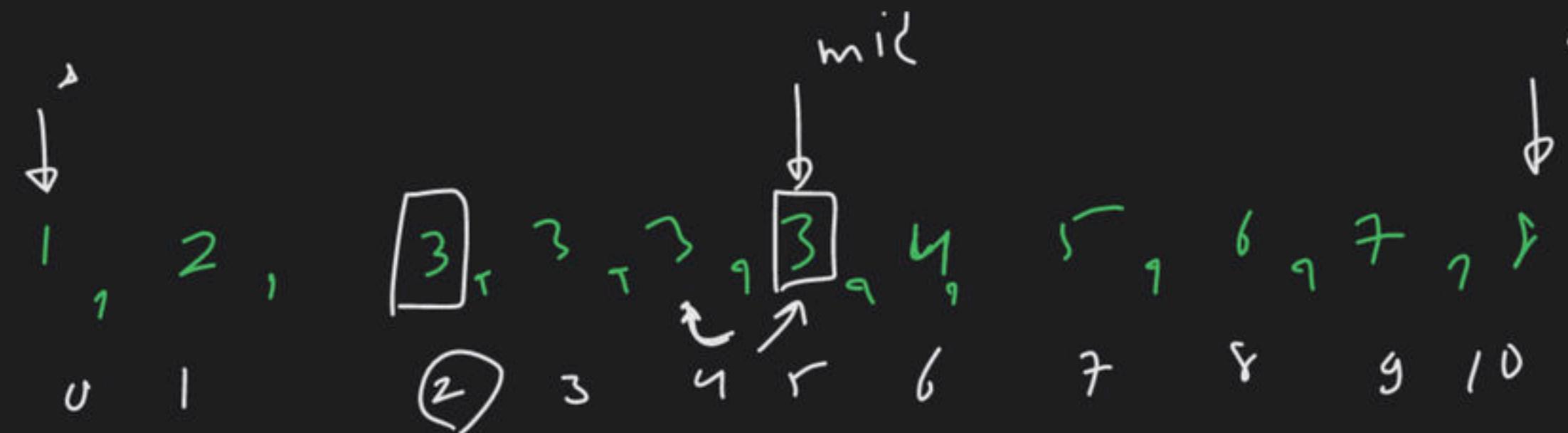
## Week-4 [Connect]

Special class



hint Occurrence

target



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

    // ans store  
    ans = mid

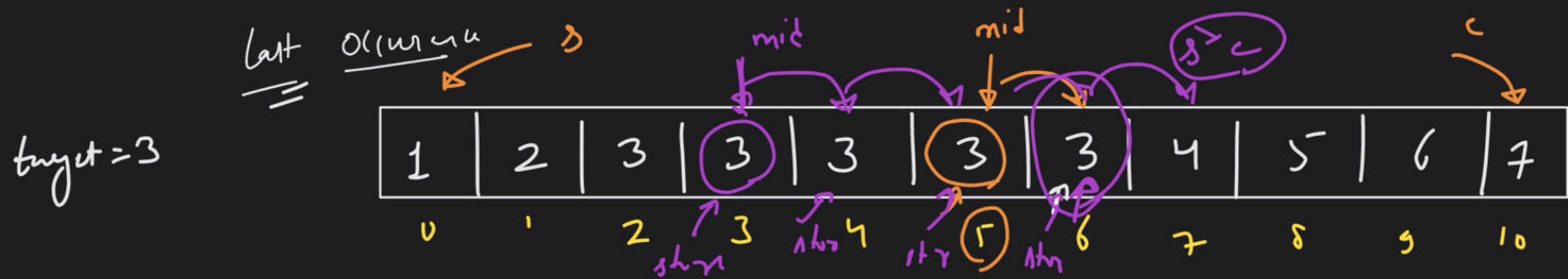
    l = s  $\rightarrow$  e = mid

if ( $arr[mid] > target$ )

    c = mid - 1;

else

    s = mid + 1;



B.S  $\rightarrow O(\log n)$   
 BS | FD  $\rightarrow O(n)$   
 VP | VD

if (arr[mid] == target)  
 (      ↳ ans store  $\rightarrow$  ans = mid;  
      ↳ Q.S  $\rightarrow$  s = mid + 1 )

Selection Sort  
 Insertion Sort  
 Bubble Sort

if (arr[mid] > target)  
 (      ↳ l.s  $\rightarrow$  c = mid - 1;  
      ↳ s = mid + 1 )

11 12 13 14 15 16 17 18

11 → 18

18 - 11 → 7 + 1

= 8

L.B  
U.B

→ STL → predefined

Total no. of Occurrence

Total =

$\lceil \log^{\frac{1}{2}} \cdot \text{occ} - \text{first occ} + 1 \rceil$

$\log n - \log(\_)$

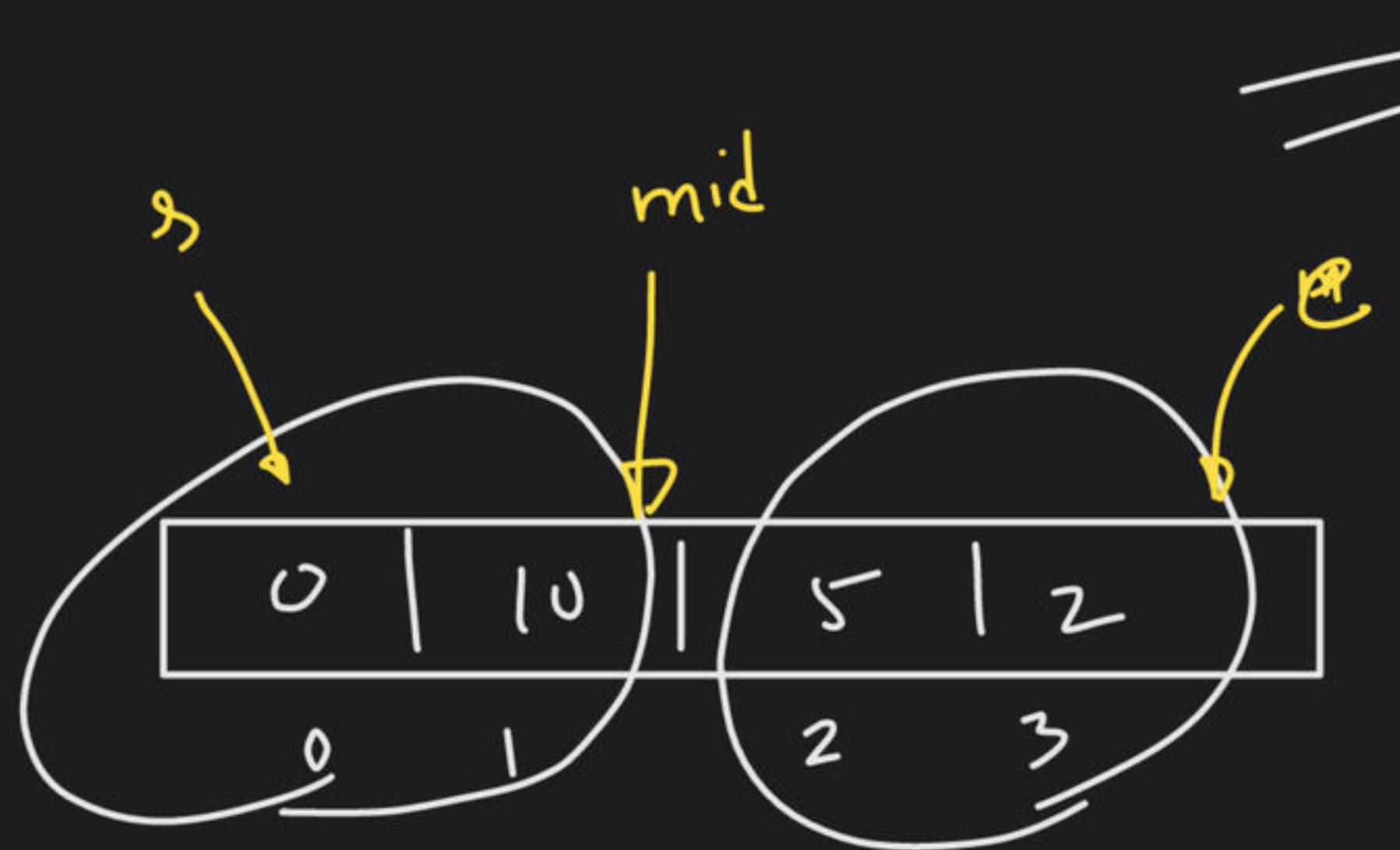
$O(\log n)$

$\frac{11 - 18}{2}$

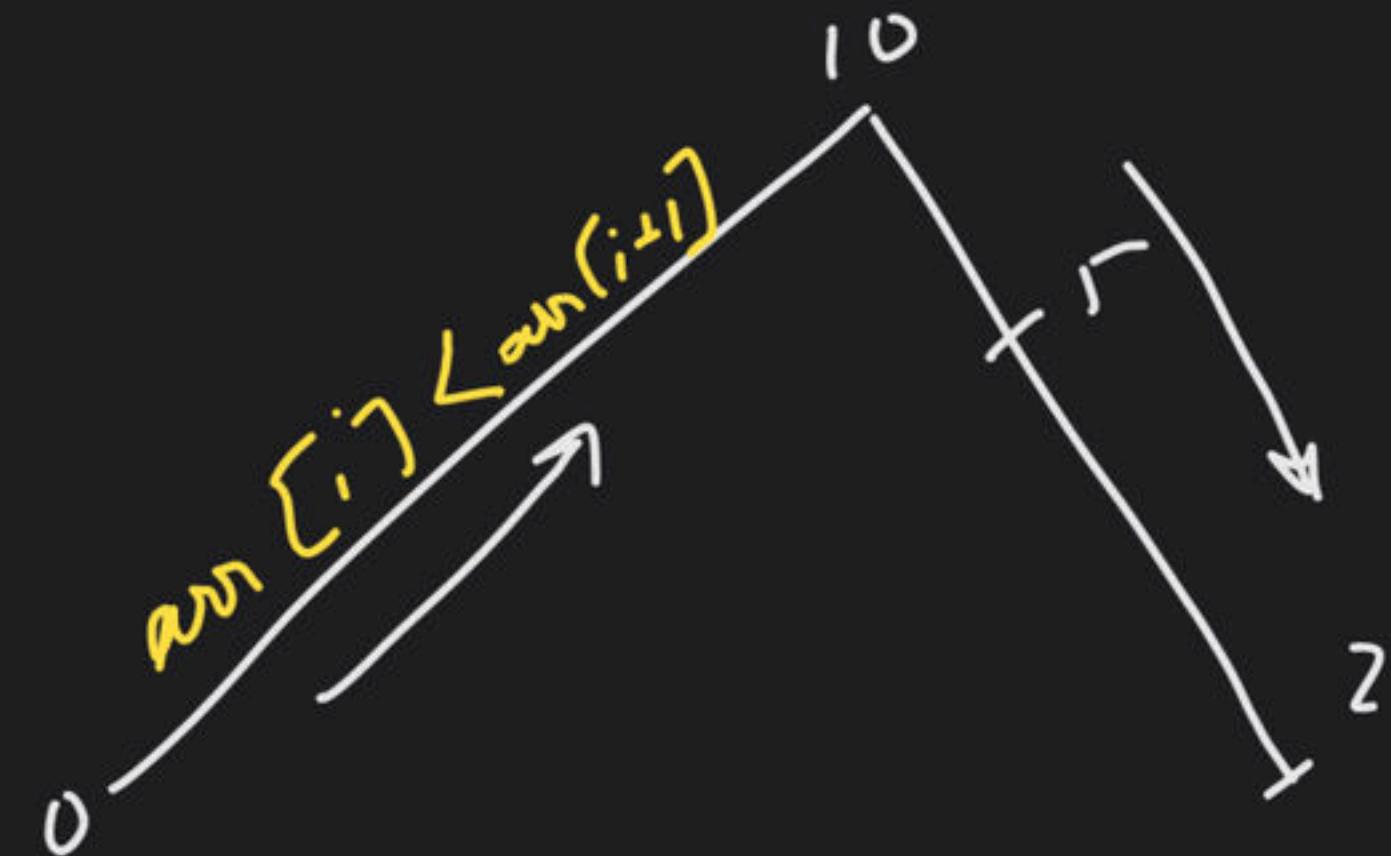
40 → 9

$18 - 11 = 7 + 1$   
= 8

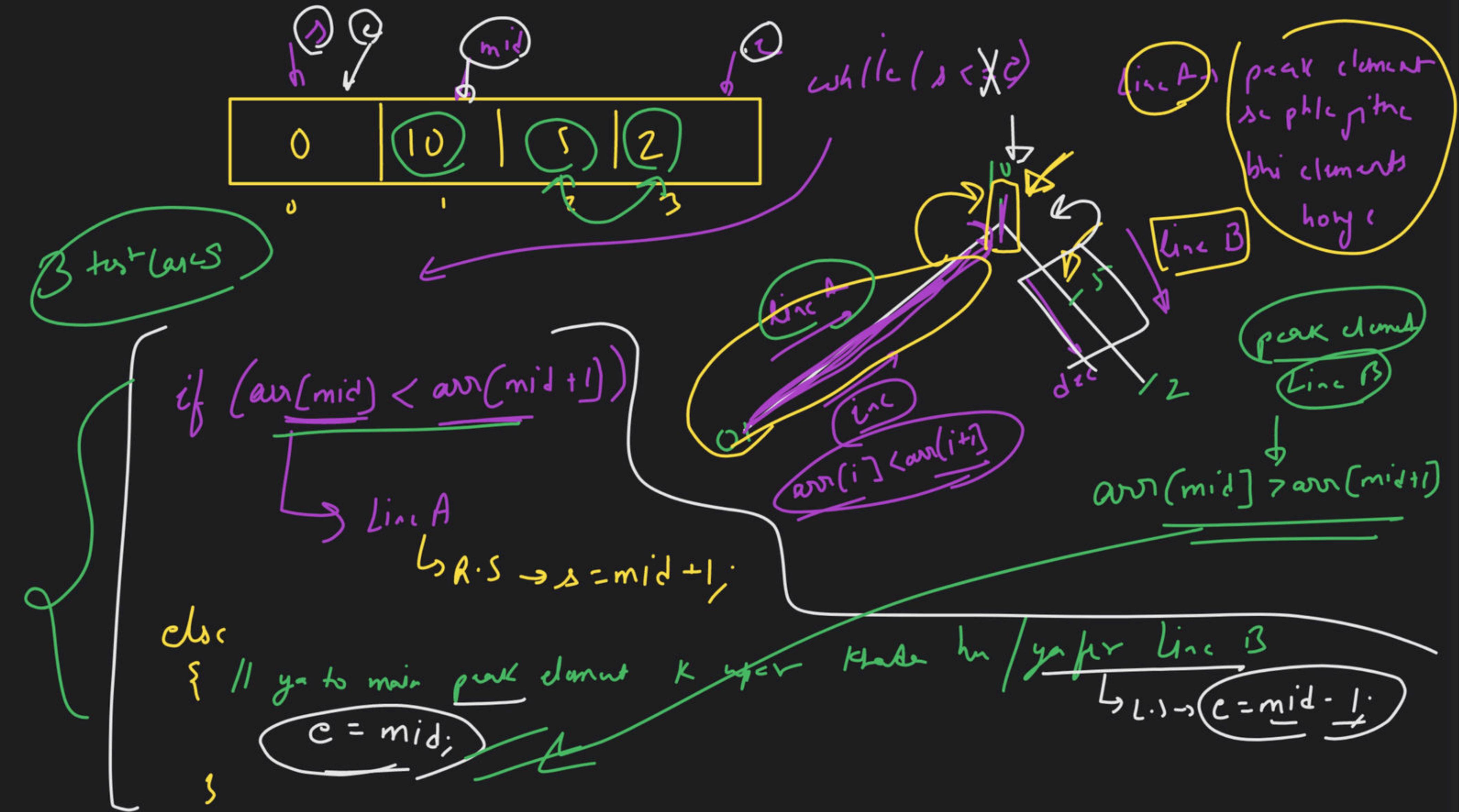
Peak Element

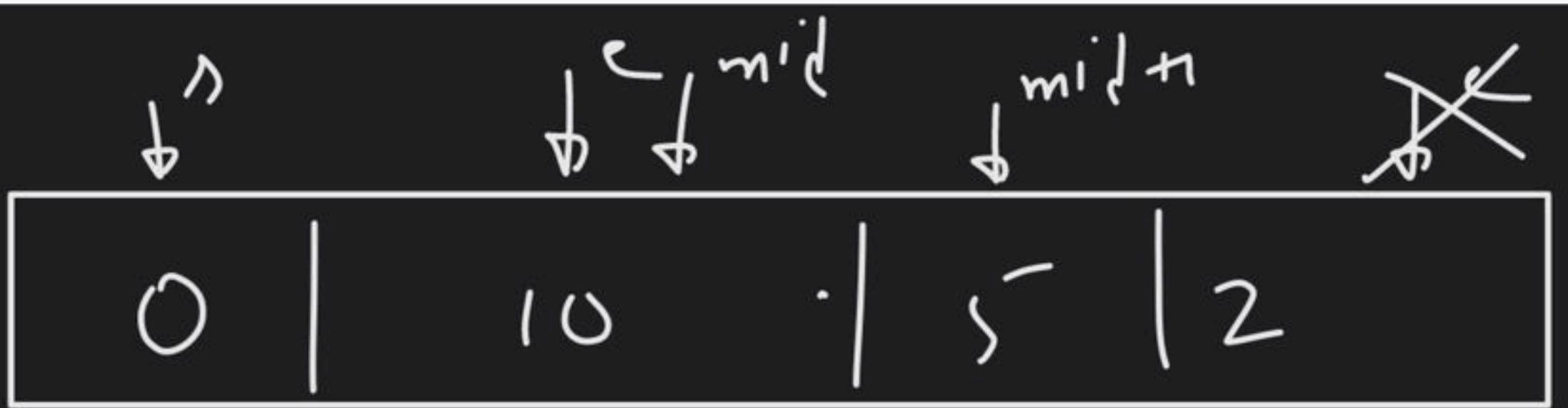


(A)



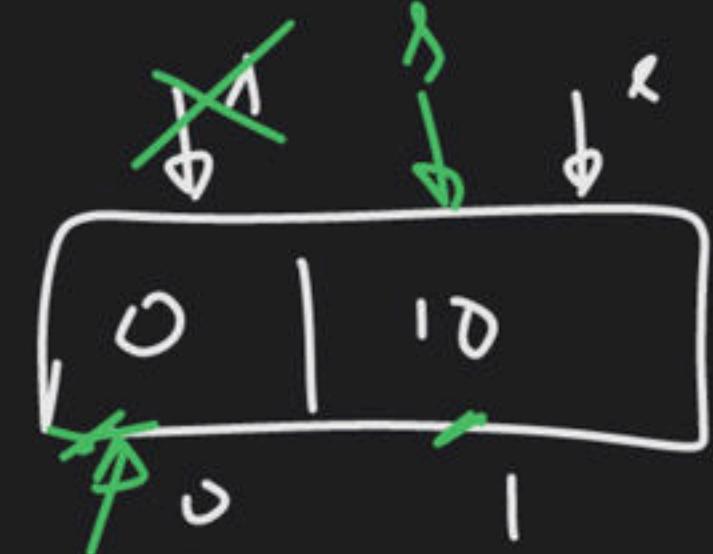
(B)



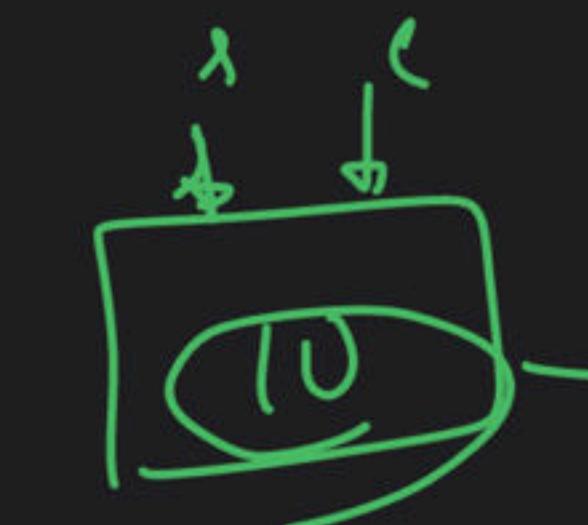


0 | 10 | . | 5 | 2

0 1 2 3



mid



return s

$0 < 10$

$\rightarrow s = mid + 1$

$s = c \rightarrow \text{loop L b l h a r}$

$10 > 5$

↓

else

$b = mid$

$(s < c)$

π<sup>hL</sup> class

↳ Pivot Element

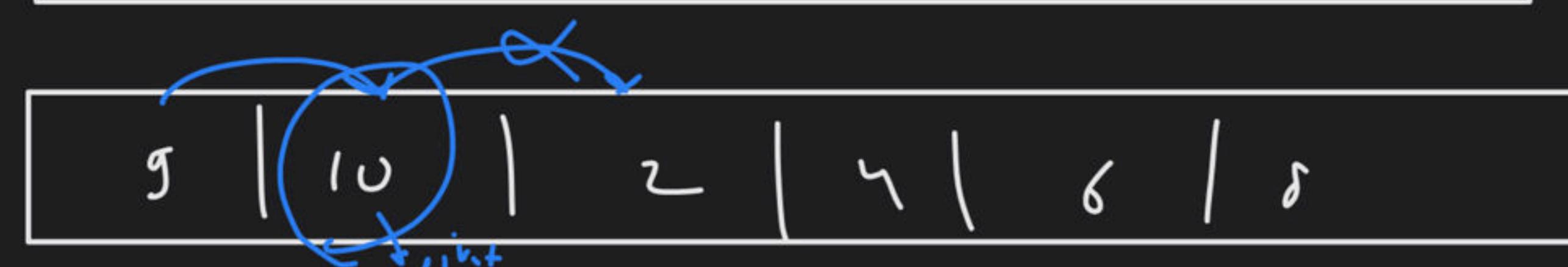
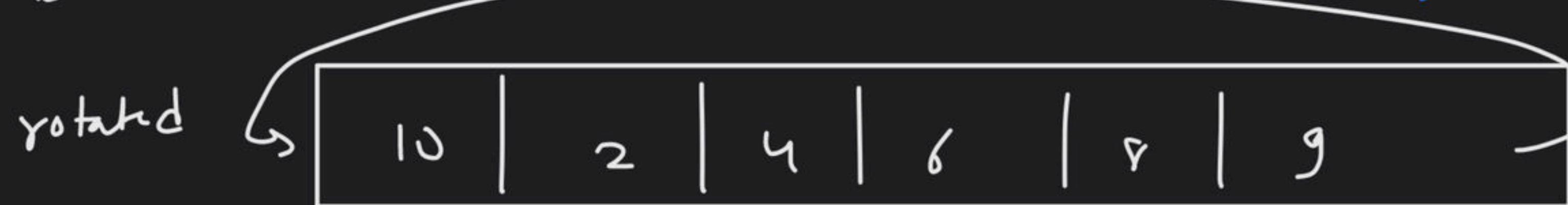
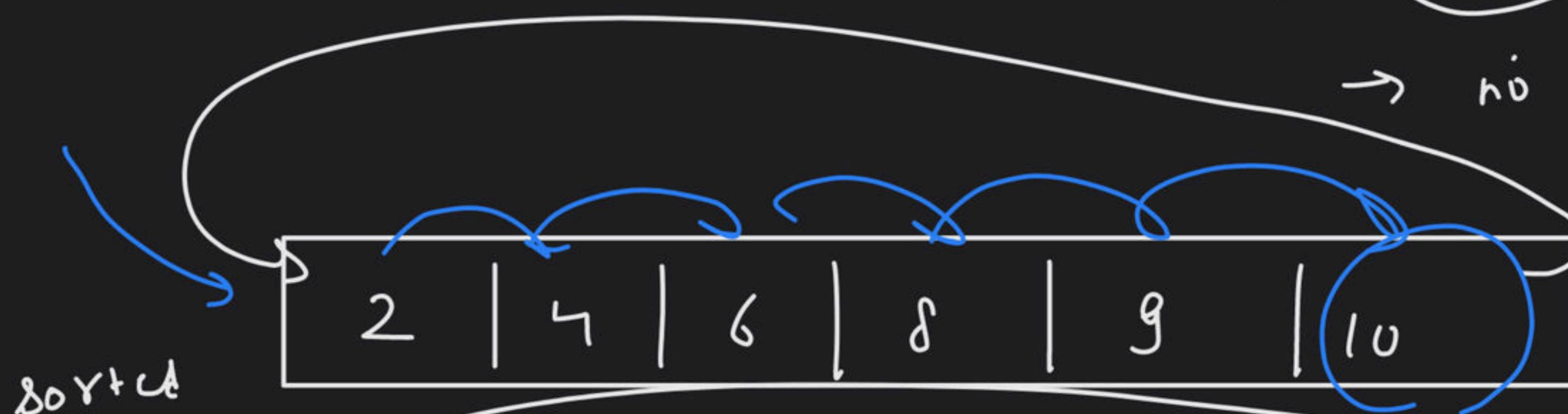
↳  $i_{lp} \rightarrow$  rotated

↳ sorted array

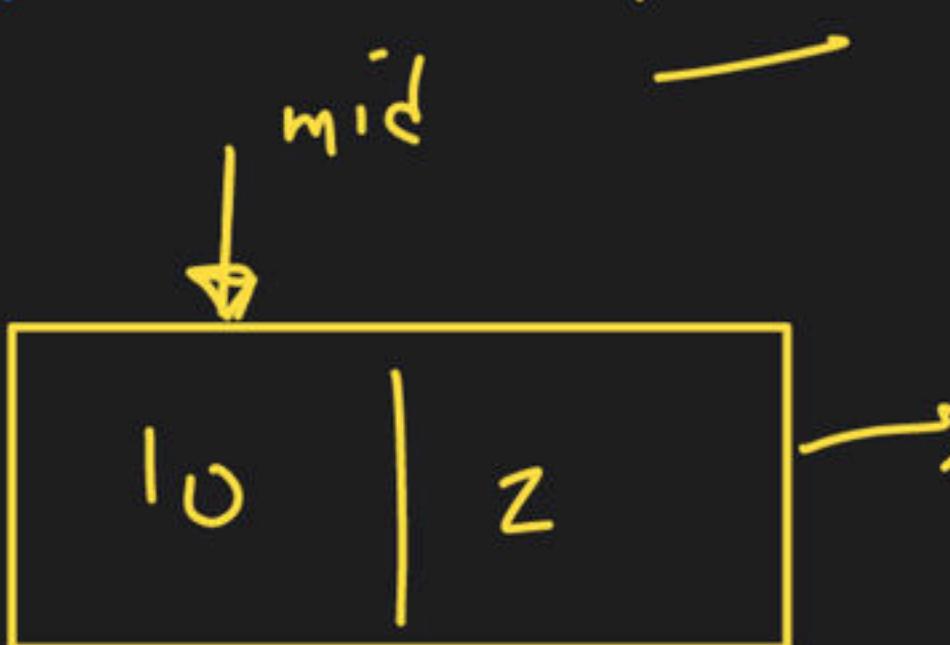
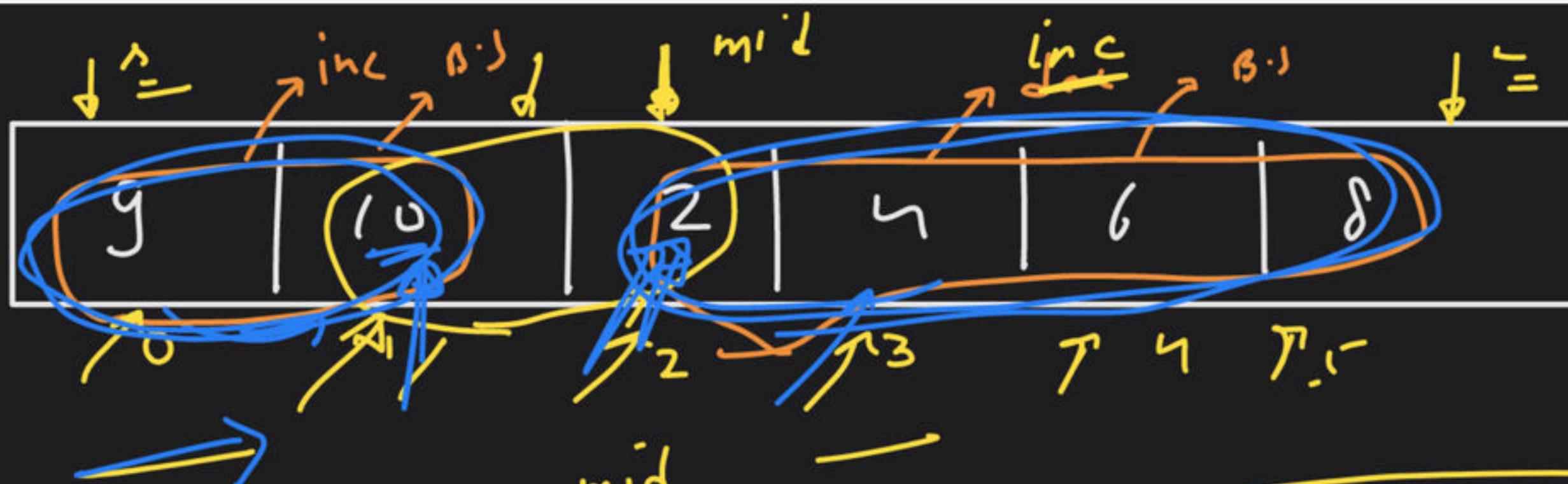
→ no

Duplicate element

~~7 7 7 7 12~~



$$\text{mid} \sim \frac{l+r}{2}, \frac{0+1}{2} = \frac{\Sigma - 3}{2}$$



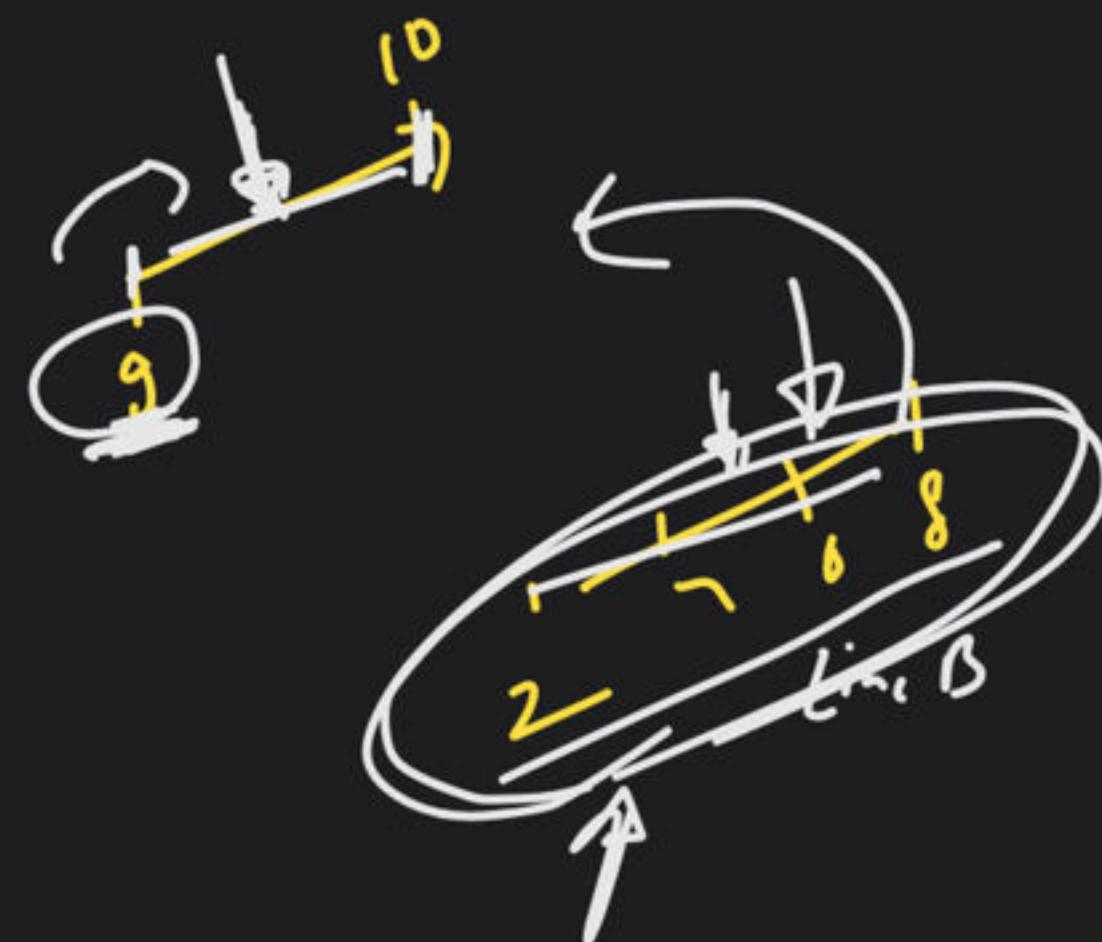
```
if (arr[mid] > arr[mid+1])
    return mid
```



```
if (arr[mid-1] > arr[mid])
    return mid - 1
```

9		10		2		4		6		8
0		1		2		3		4		5

if ( $\text{arr}[s] > \text{arr}[\text{mid}]$ )  
 $\hookrightarrow \text{mid} \rightarrow \underline{\text{line } B}$   
 $\hookrightarrow \text{left} \leftarrow c = \text{mid} - 1$   
 else  
 $s = \text{mid} + 1$





```
int s = 0;  
int e = n - 1;  
int mid = s + (e - s) / 2;  
while (s <= e) {  
    if (mid + 1 <= e && arr[mid] > arr[mid + 1])  
        return mid;  
    if (arr[s] > arr[mid])  
        s = mid + 1;  
    else  
        e = mid - 1;  
}
```

↓ duplicate  
element

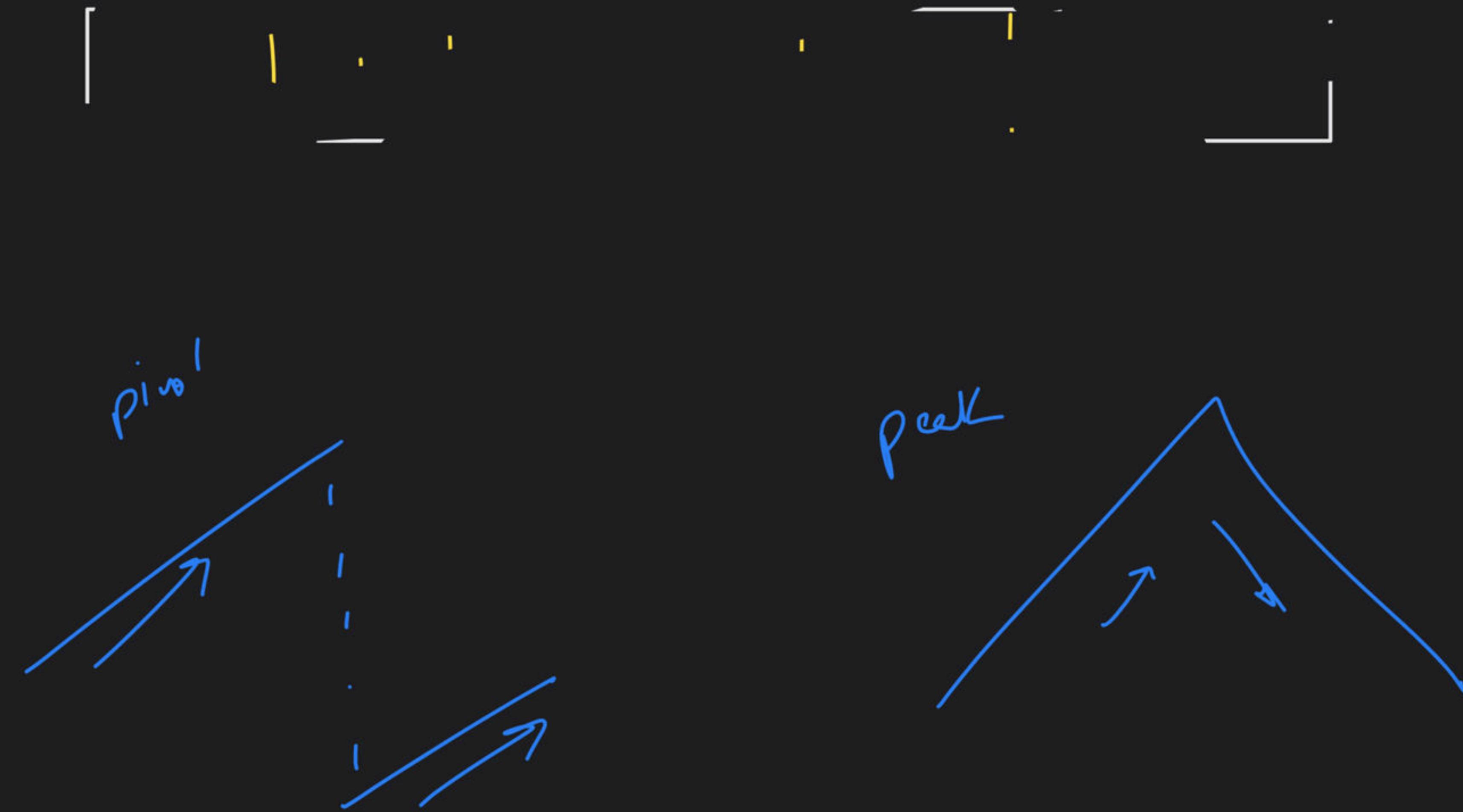
↓ no rotation

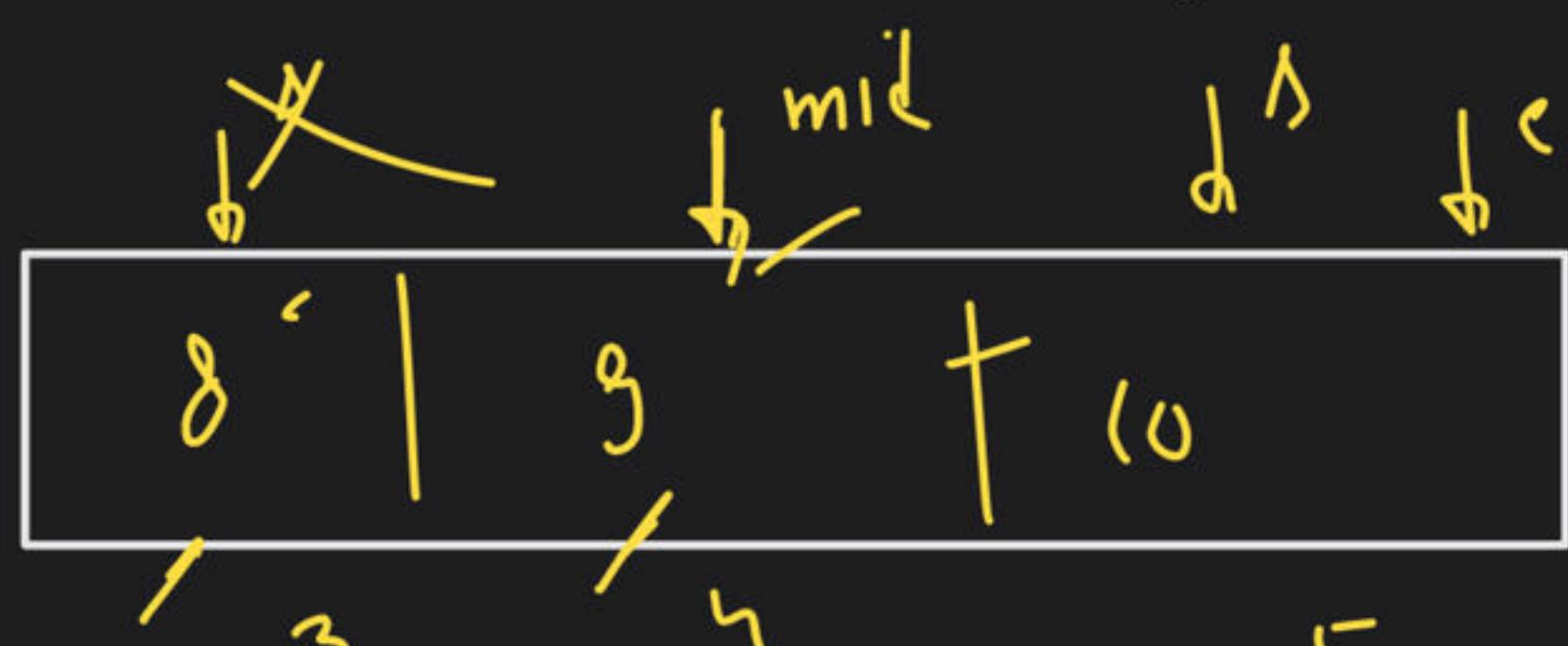
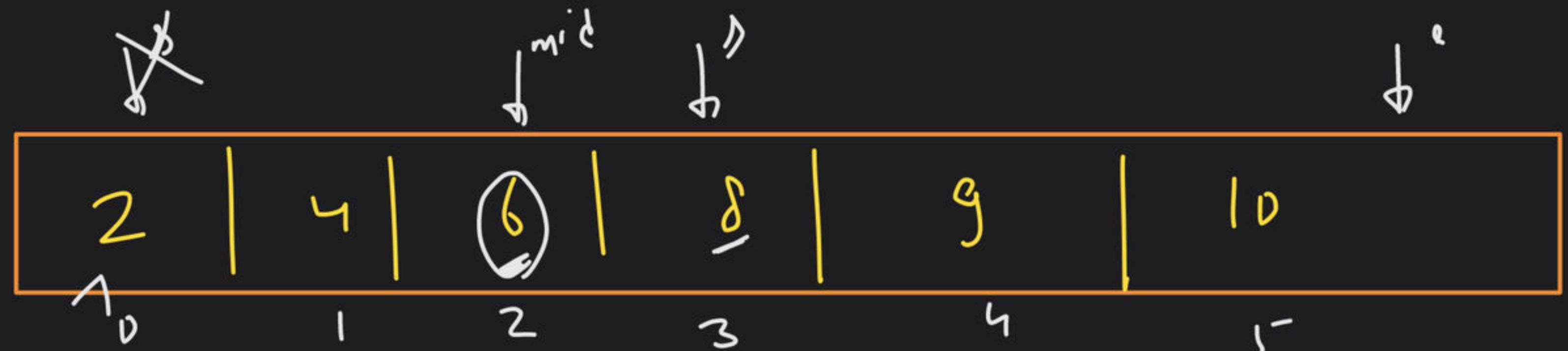
if (mid - 1 >= s && arr[mid - 1] > arr[mid])  
 return mid - 1;  
if (arr[s] > arr[mid])  
 s = mid + 1;  
else  
 e = mid - 1;

mid > end  
you

mid = s + (e - s) / 2;

3 → return -1

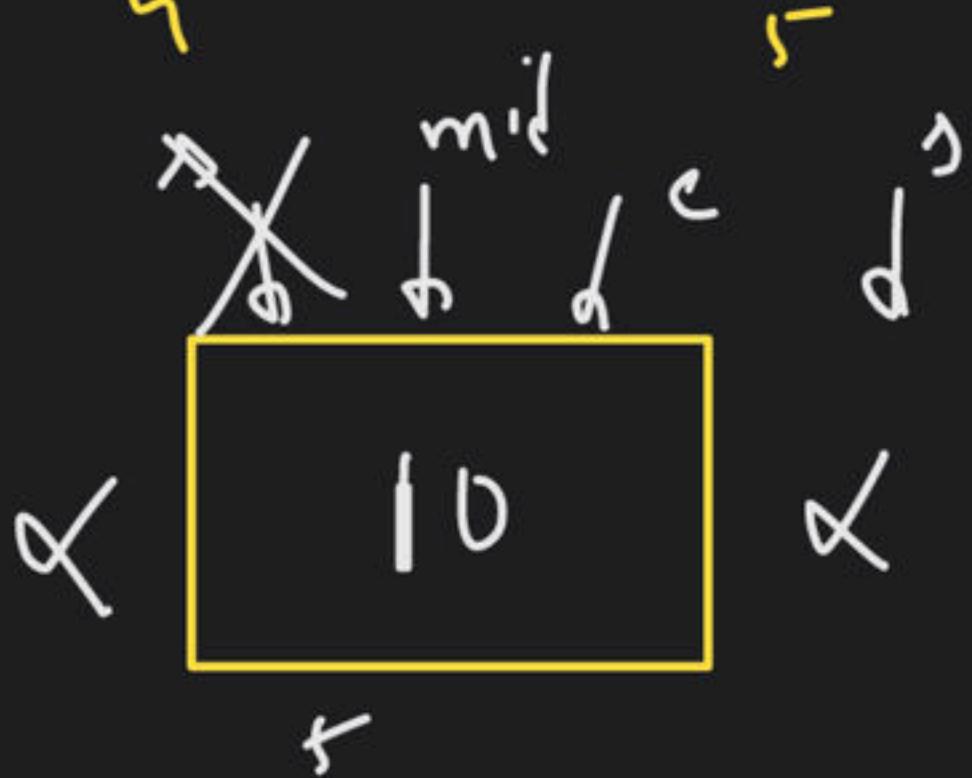




$\delta = 6$   
 $c = 8$

loop  
break

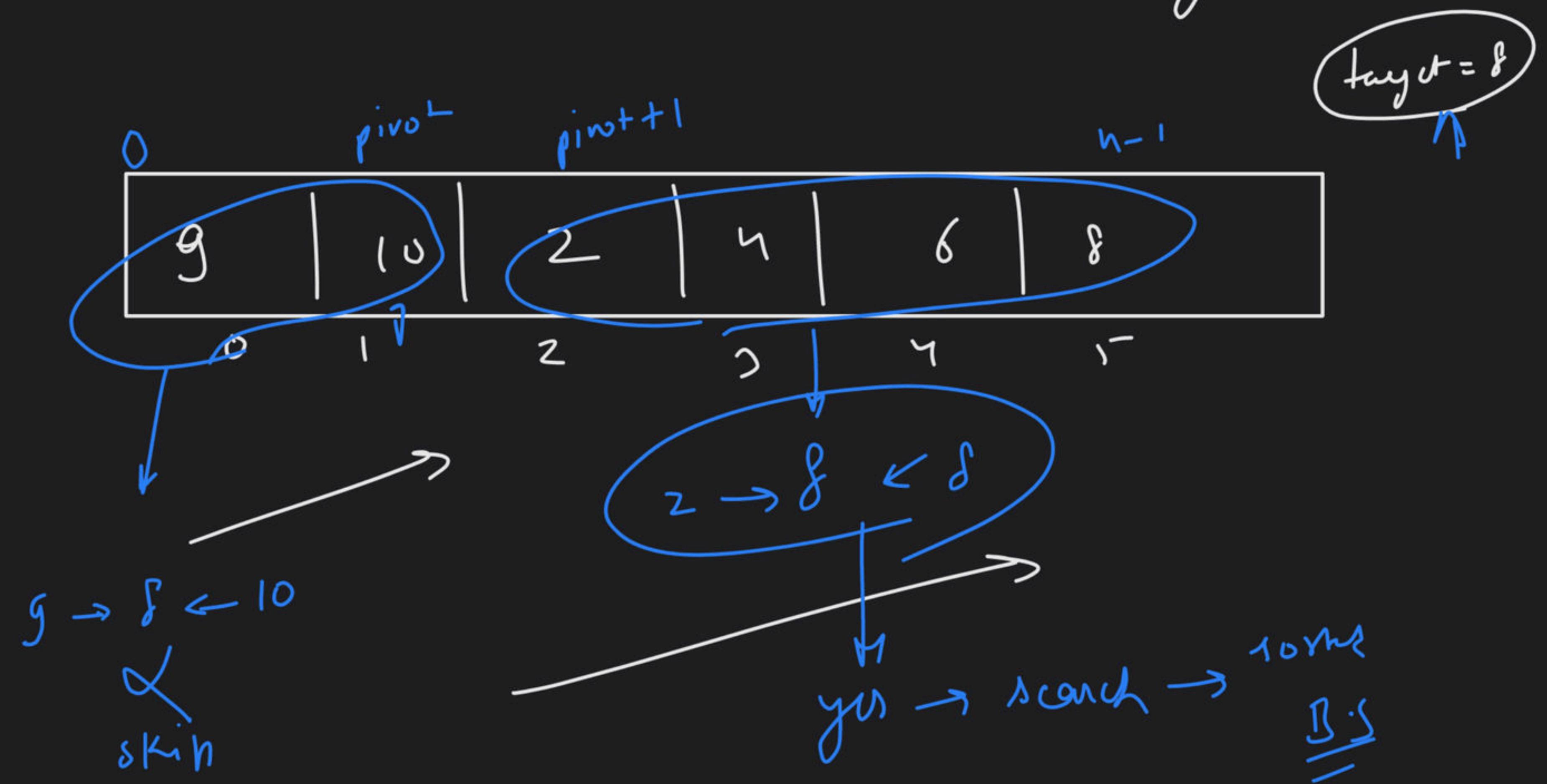
return -1



$10 > 10$

X

search in a ~~sorted~~ rotated & sorted array





$$\xrightarrow{\quad} \underline{\underline{S_{\theta} r^+}}$$

$$i/p \rightarrow n = 25^-$$

$$i/p \rightarrow a\omega = \sqrt{25^-} = 5$$

$$\sqrt{59}$$

$$= 9 \cdot \text{circle}$$

$$29$$

$$\sqrt{27} \rightarrow \boxed{5} \text{ circle}$$

$$\sqrt{25} \rightarrow \boxed{5} \sim$$

$$\sqrt{39} \rightarrow \boxed{6} \cancel{5}$$

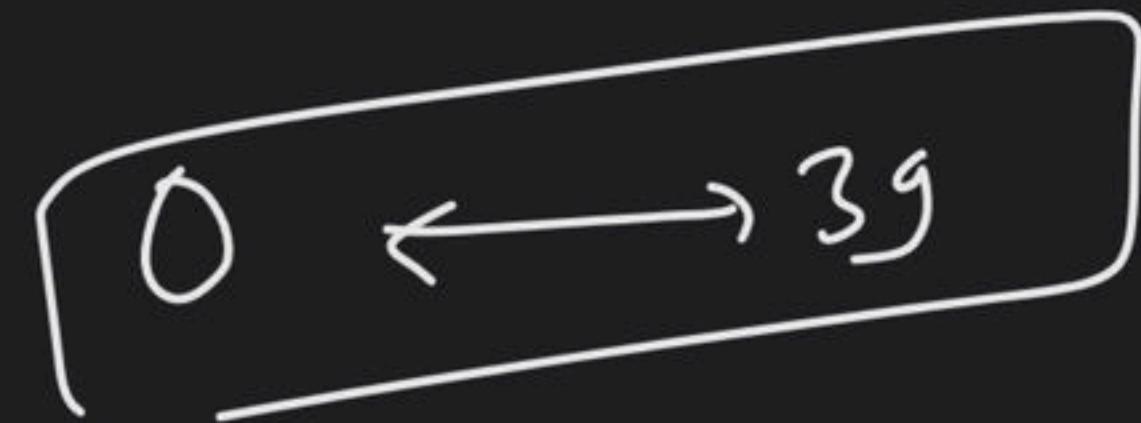
$$\sqrt{101} \rightarrow \boxed{10} \cancel{5} \rightarrow 10$$



$$\sqrt{25} \rightarrow \text{ans}$$



$$\sqrt{35} \rightarrow \text{ans}$$



$$\sqrt{101} \rightarrow \text{ans}$$



$$17 \times 17 = 289 \neq 35$$

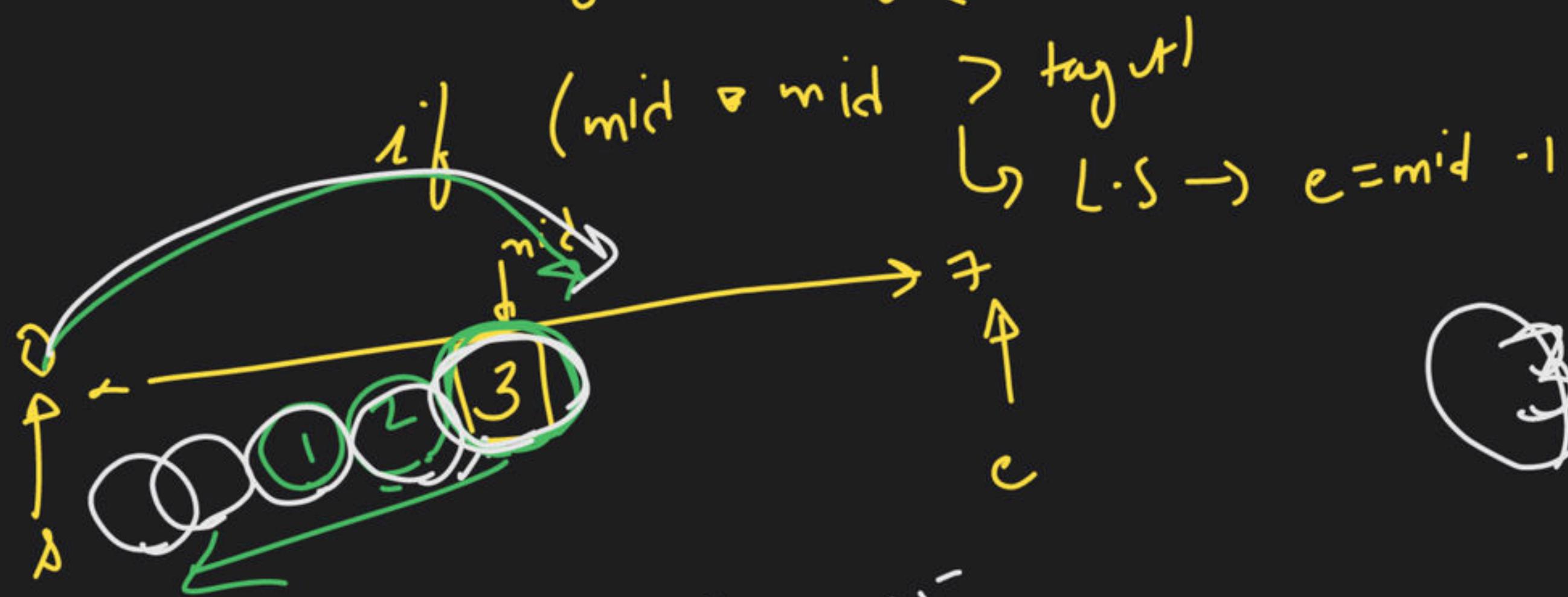
$$\sqrt{35} \rightarrow \text{ans}$$



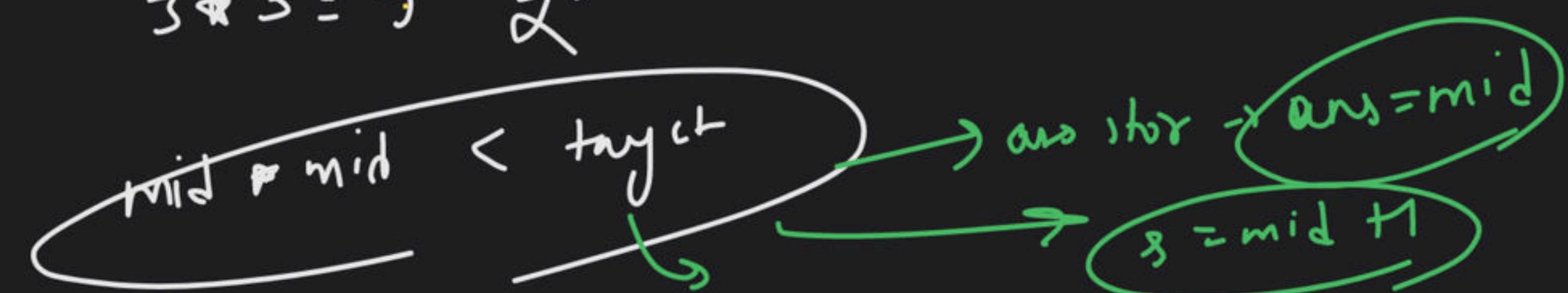
mid \* mid == target  $\rightarrow L.S \rightarrow c = mid - 1$



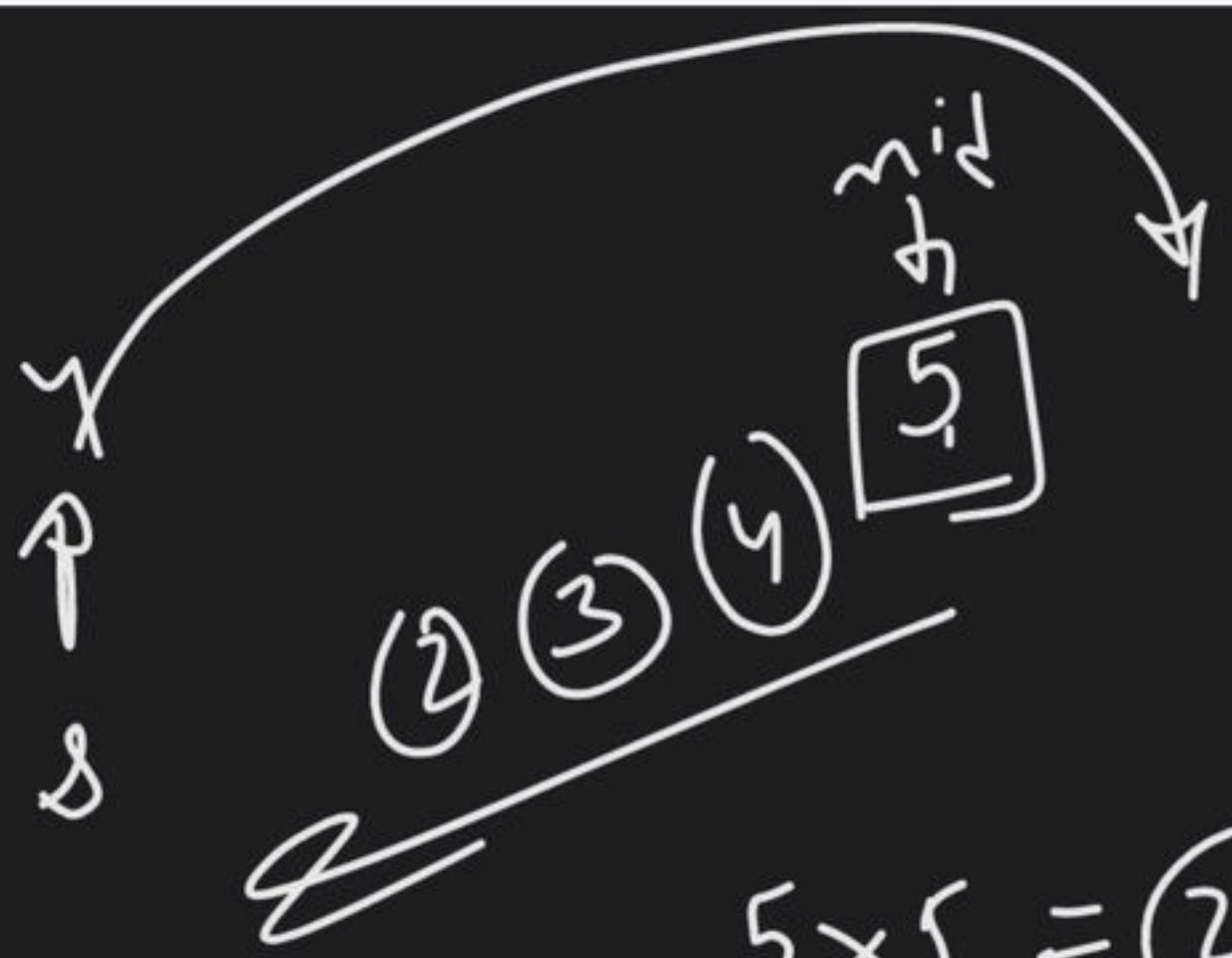
$$8 \times 8 = 64 \cancel{<} 35$$



$$3 * 3 = 9 \cancel{<} 35$$

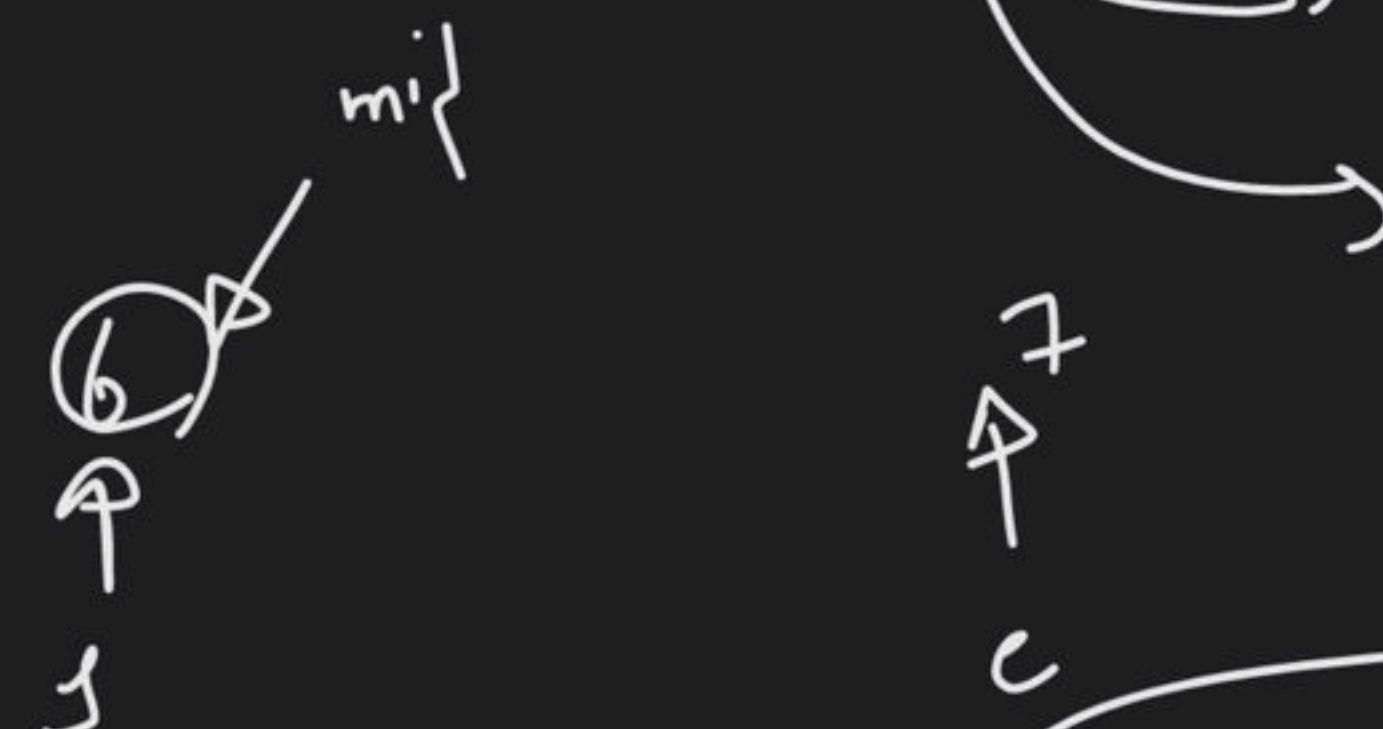


~~mid~~



$$5 \times 5 = 25 < 35$$

$$\text{ans} = \frac{\text{ans} + 1}{\sqrt{35}}$$



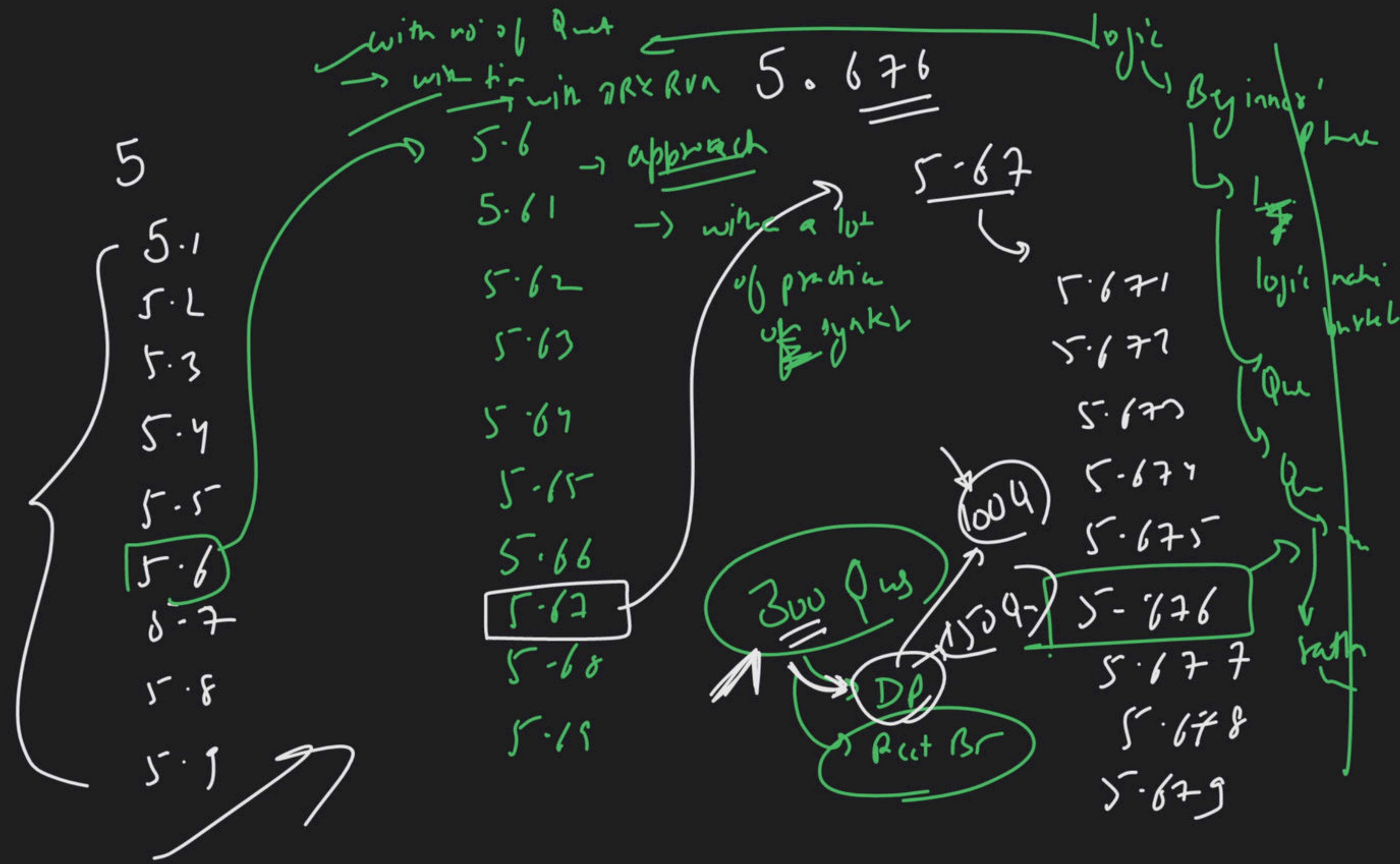
$$6 \times 6 = 36 \geq 35 \rightarrow c = mid - 1$$
$$c = 5$$



$\delta = 6$

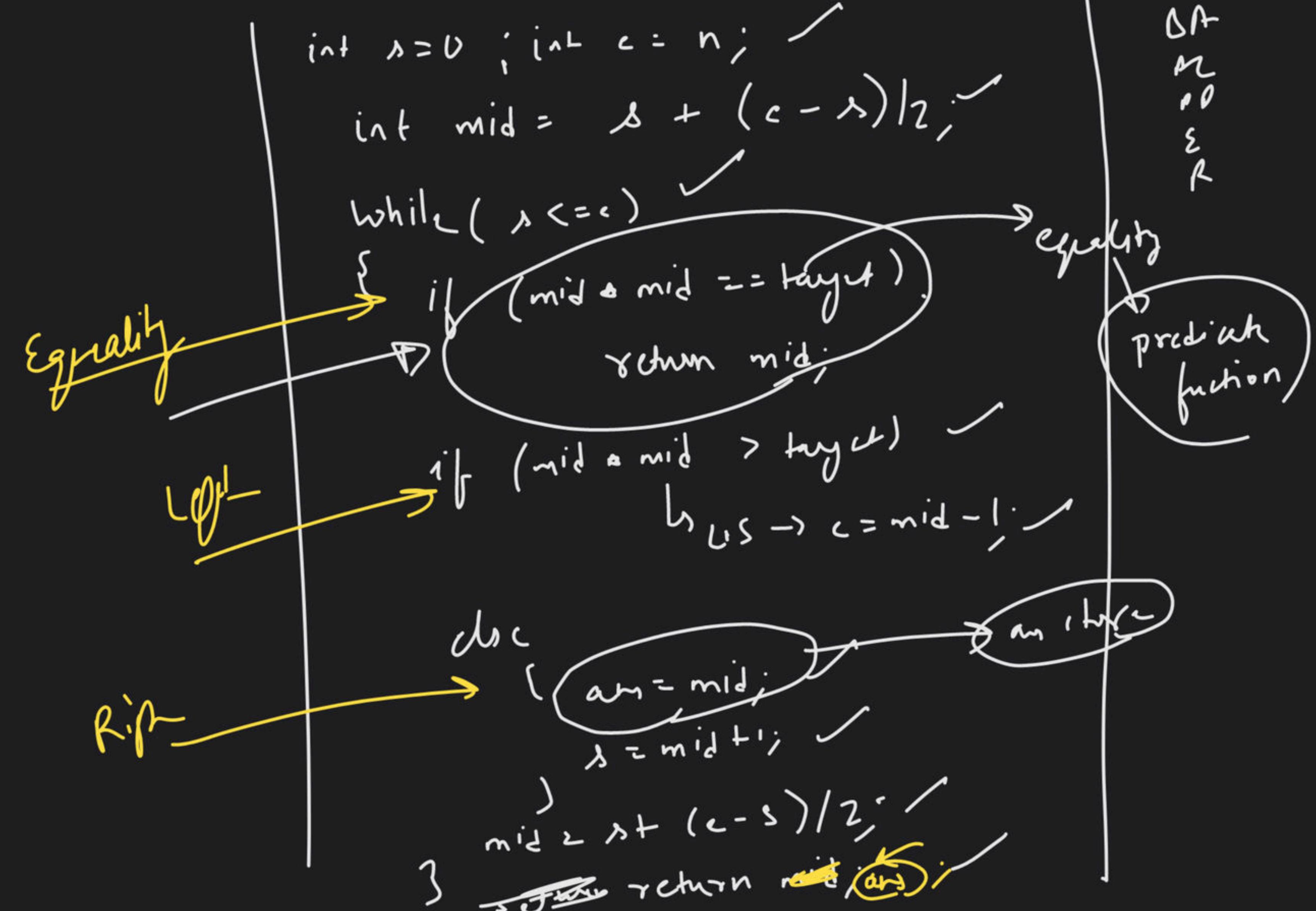
$c \approx 2$

$\delta > c \rightarrow$  loop stop

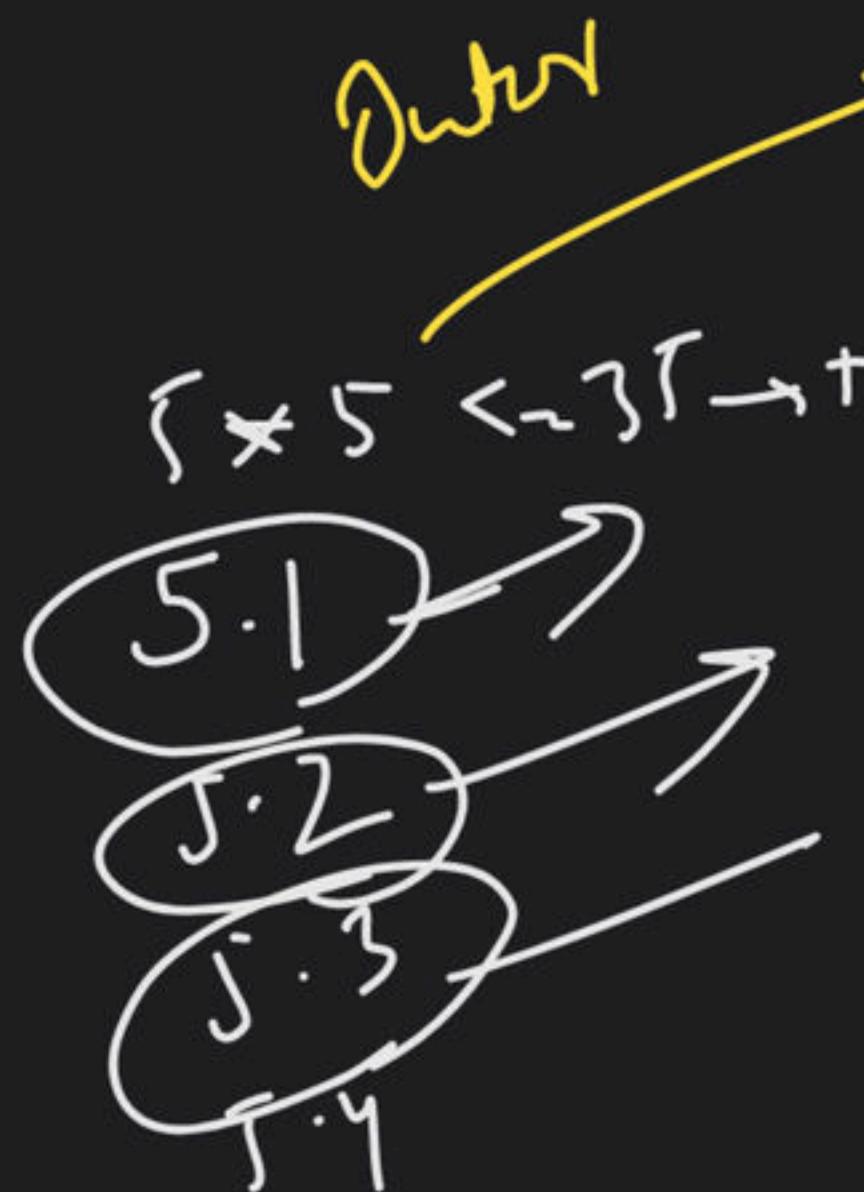


DATA  
N  
P  
E  
R

Search  
Space



5.  76



0.01

}

double step = 0.1;

for ( i=1 ; i <= precision ; i++ )

{

for ( double j = finalAns ;  
j \* j <= target ; j = j + step )

{

. finalAns = j ;

}

$$\begin{array}{r} 5 \cdot 6 \\ 0 \cdot 01 \\ \hline 5 \cdot 61 \\ 5 \cdot 62 \end{array}$$

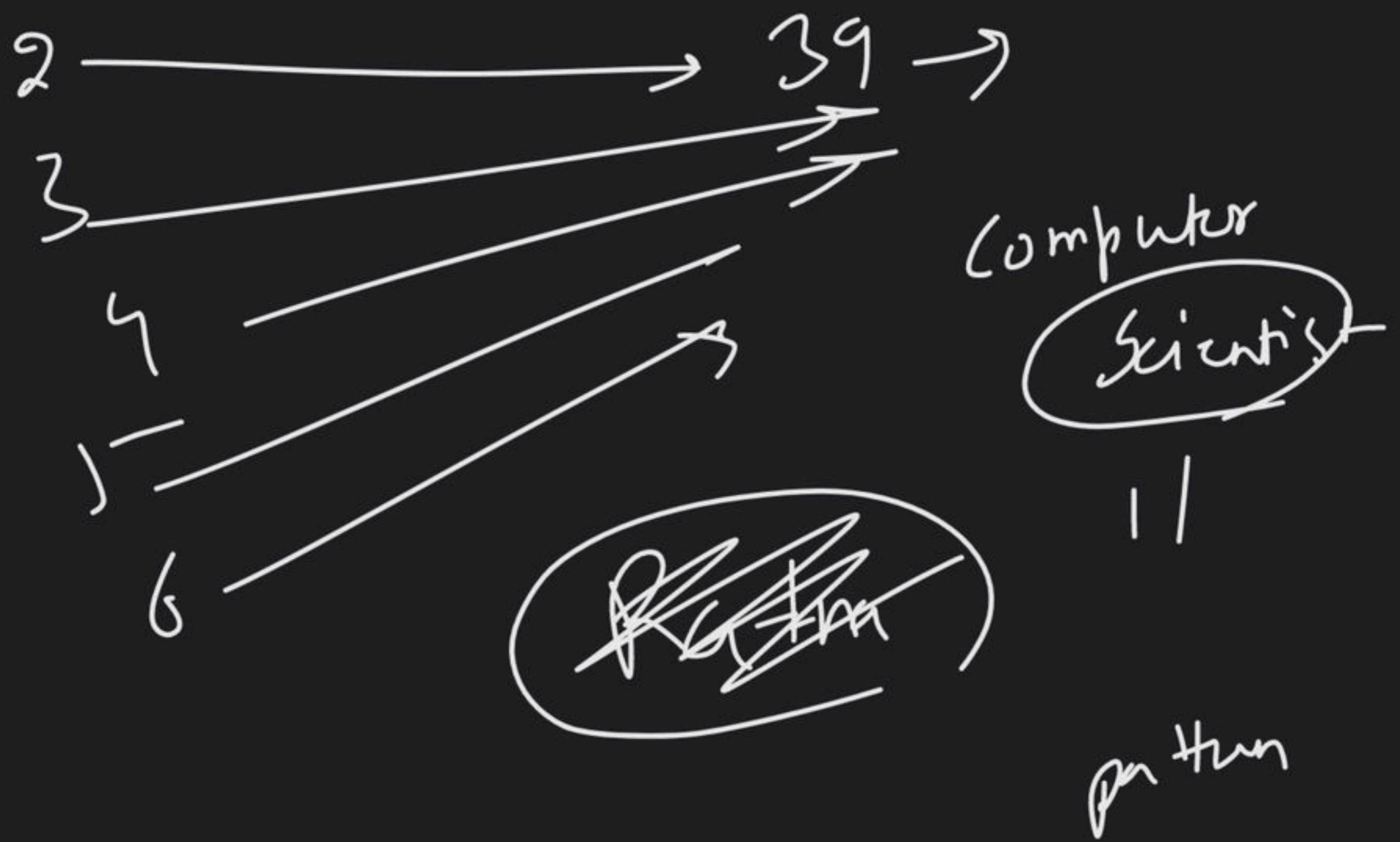
step = step / 10 ;

}

5.1  
5.12  
5.123  
5.1234  
fehler -> loop

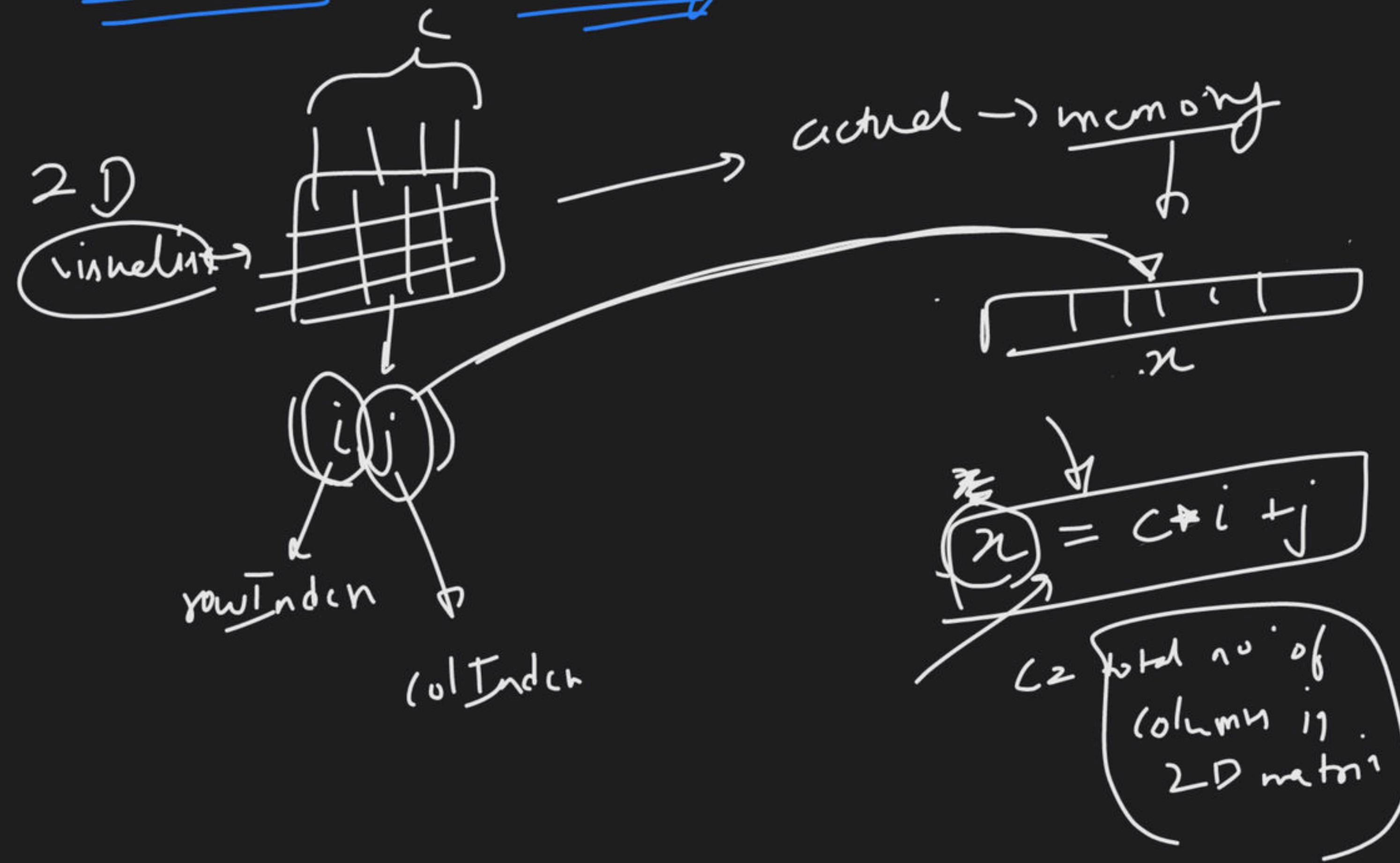
sk. 5.6

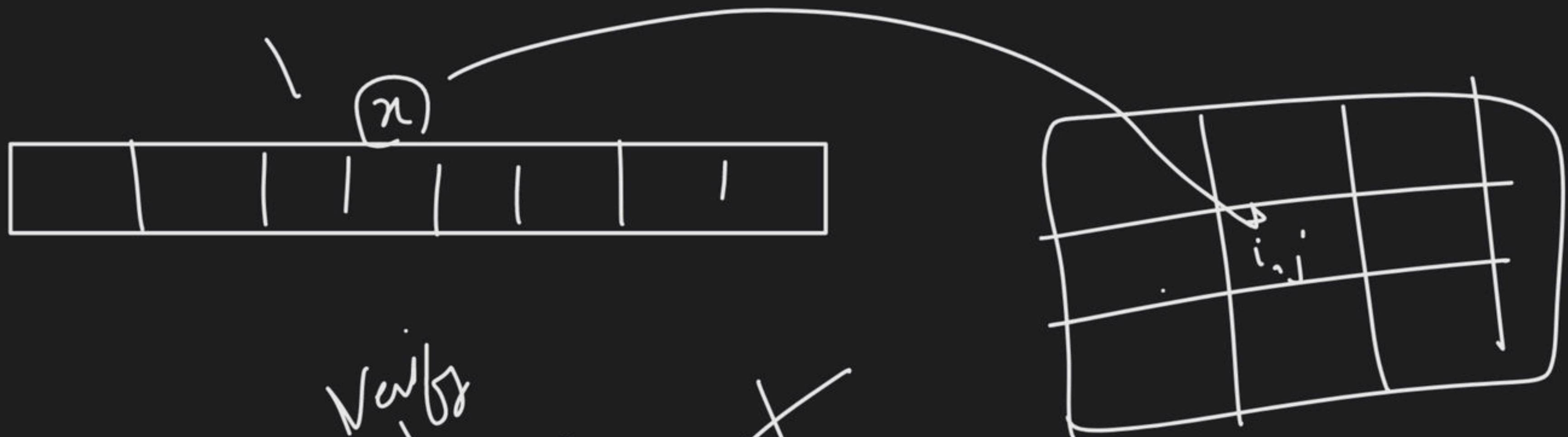
double finalAns = ans ;



C  
X.

$\Rightarrow$  3.5 in 2D array





Do my  
Run

Walls

$$i = \frac{n}{c}$$

$$j = \frac{n - loc}{mid \cdot c}$$

mid  
c

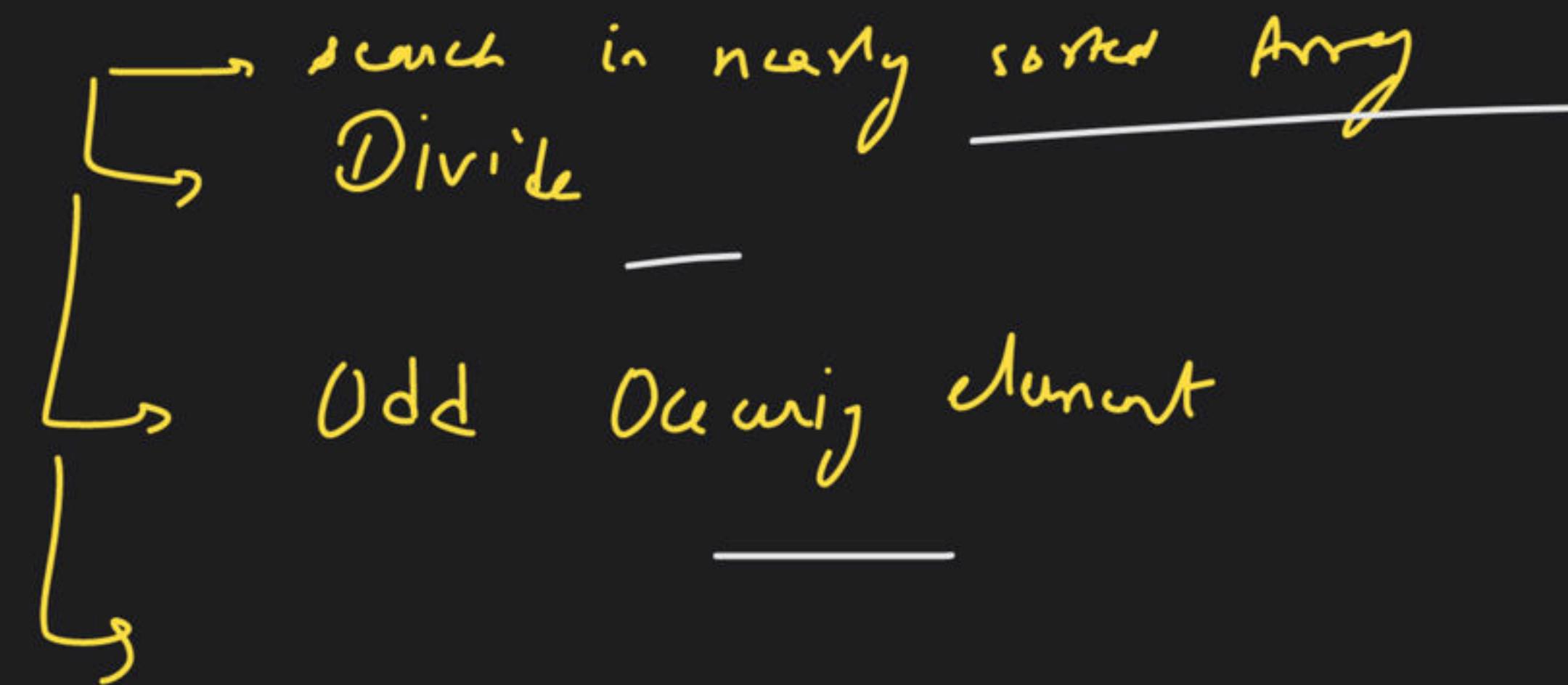
mid

1  
c

mid =  $\left(\frac{n - loc}{mid \cdot c}\right)$

$j + (c - s)/2$

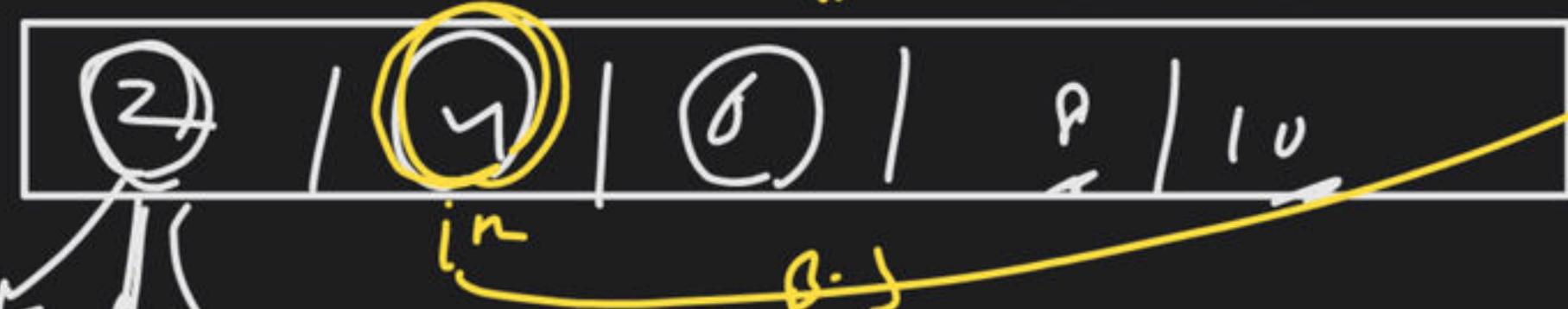
$\rightarrow$  B's Lecture - 3



→ Search in a nearly sorted Array

do the array

mid



i i+1 mid

if (Equality)  $\rightarrow$  arr[mid] == target

L-S

R-S

(arr[mid] == target)  
return mid

if (arr[mid-1] == target)  
return mid-1

(arr[mid+1] == target)  
return mid+1

$i = mid + 1$   
 $e = mid - 2$



Divide

divisor = 2

dividend = 10

Q  
R  
=

dividend = 10

Quotient = 5

Quotient

$2 \sqrt{10}$

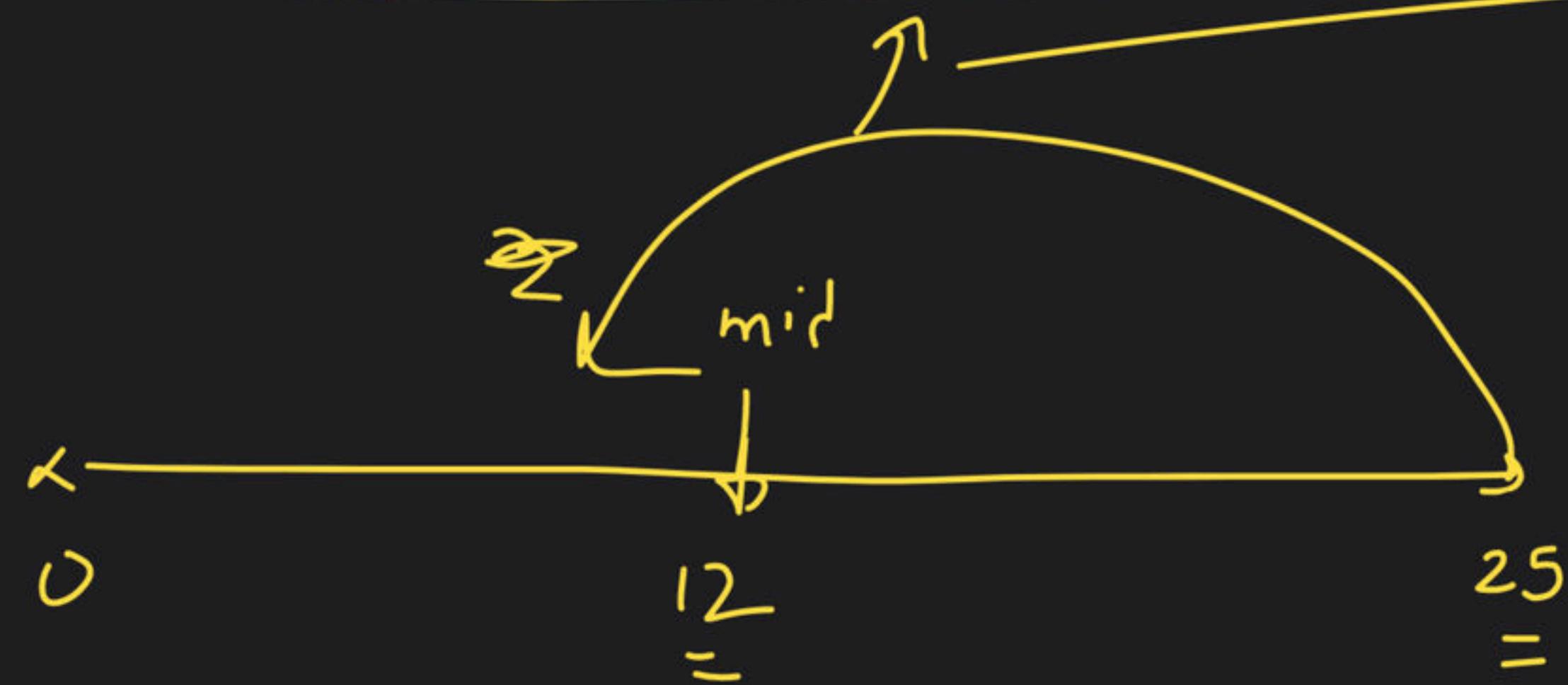
$\frac{10}{2} < \text{remainder}$

$$2 \times 5 = 10$$

$$2 \times 5 + 1 = 11$$

divisor  $\times$  Quotient + Remainder = Dividend

divisor  $\times$  Quotient  $\leq$  Dividend

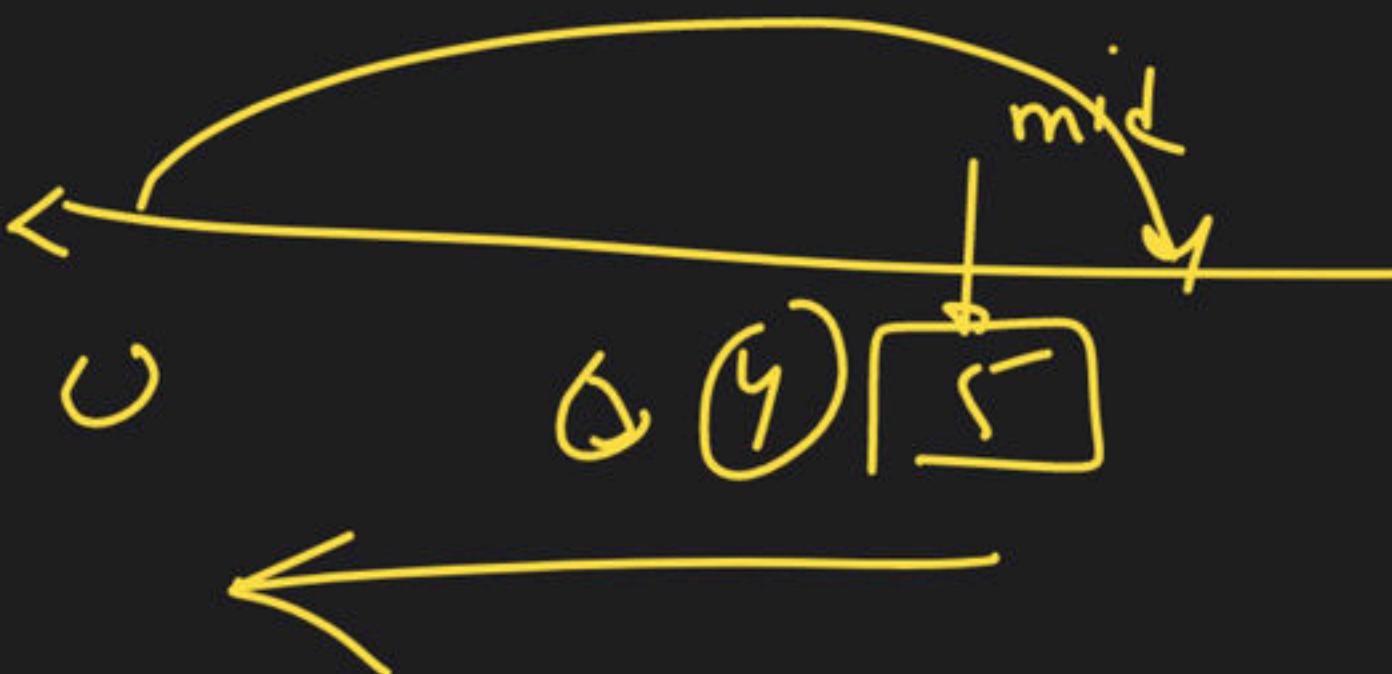


$$3 \sqrt{25} = 8$$

(B)

$$12 \times 3 = 36 > 25 \times$$

mid \* divisor > target  
↳ L.S



$\frac{25}{3} \rightarrow 8$

$$5 \times 3 = 15 < 25$$

↙ answer  
s = mid + 1



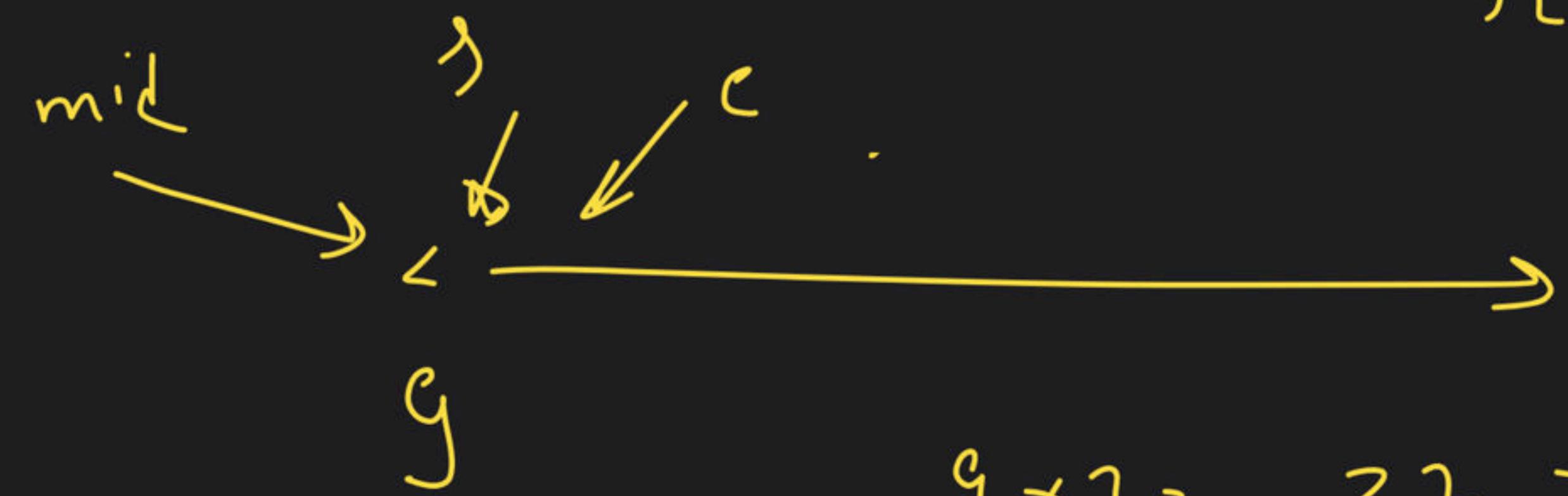
$$8 \times 3 = 24 < 25$$

↖ as store  
s = mid + 1



$$10 \times 3 = 30 > 25$$

L.S



$$9 \times 3 = 27 > 25$$

R.S

~~e = 8~~

$$\begin{aligned} e &= 8 \\ s &= 9 \end{aligned}$$

$s > c \rightarrow \text{loop back}$

→ Odd Occurring Element

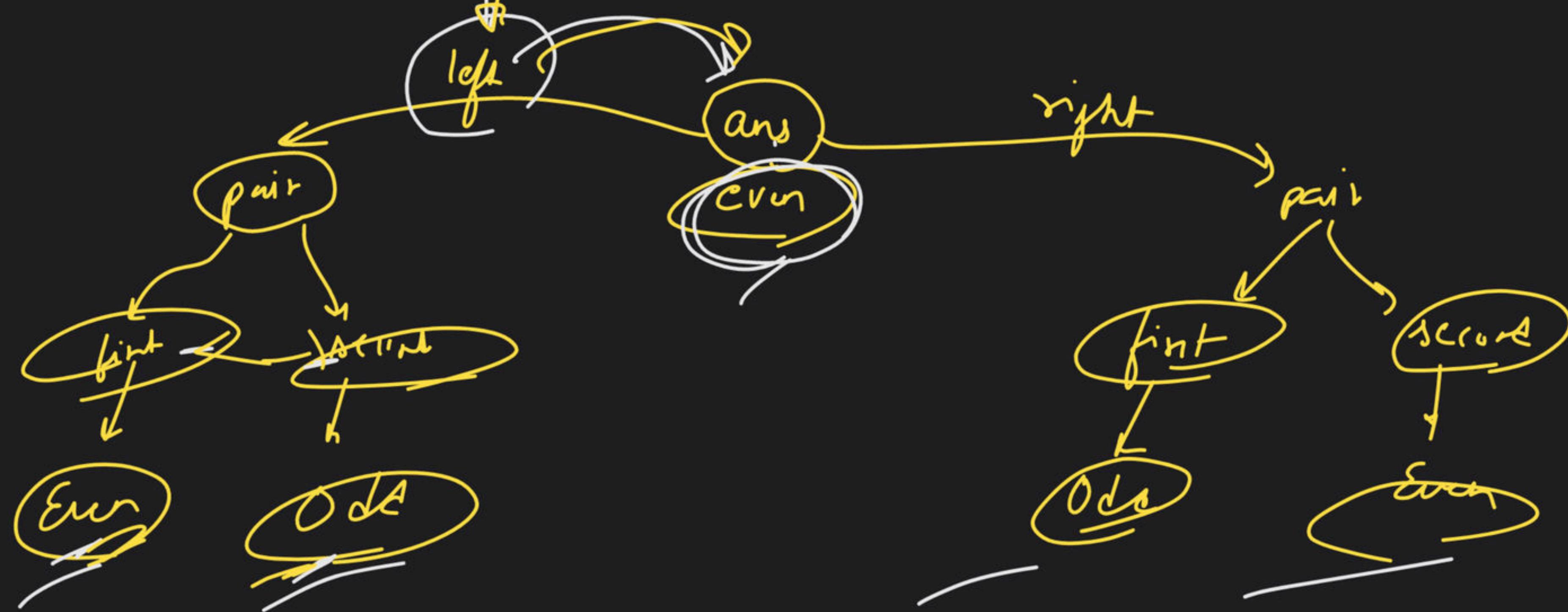
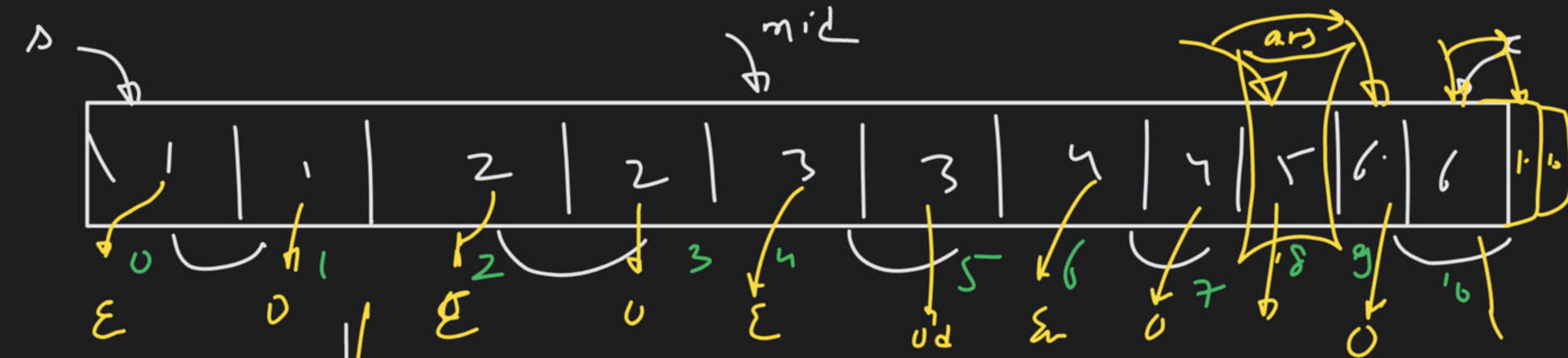
↳ array → hr element → even  
| element → odd

hr element → pairs  $(1, 1)$   $(2, 2)$   
 $(3, 3)$

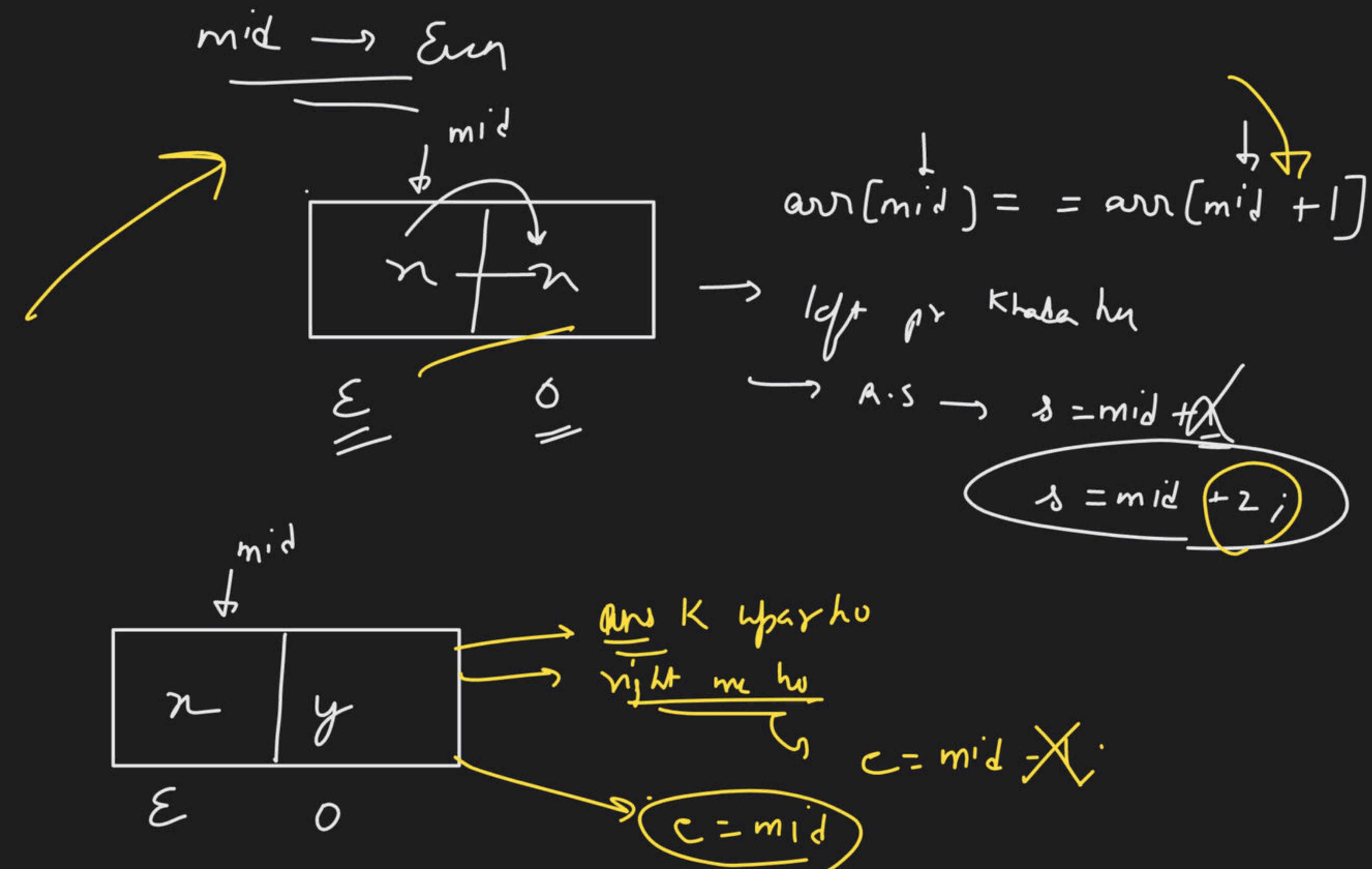
~~2 pairs~~

hr → adjacent 2 count

$\begin{matrix} 1 & 1 \\ \swarrow & \searrow \end{matrix}$   $\begin{matrix} 1 & 1 & 1 \\ \diagup & \diagdown \end{matrix}$



mid  
    
even      odd



(A)

if ( $mid \% 2 == 0$ )

{

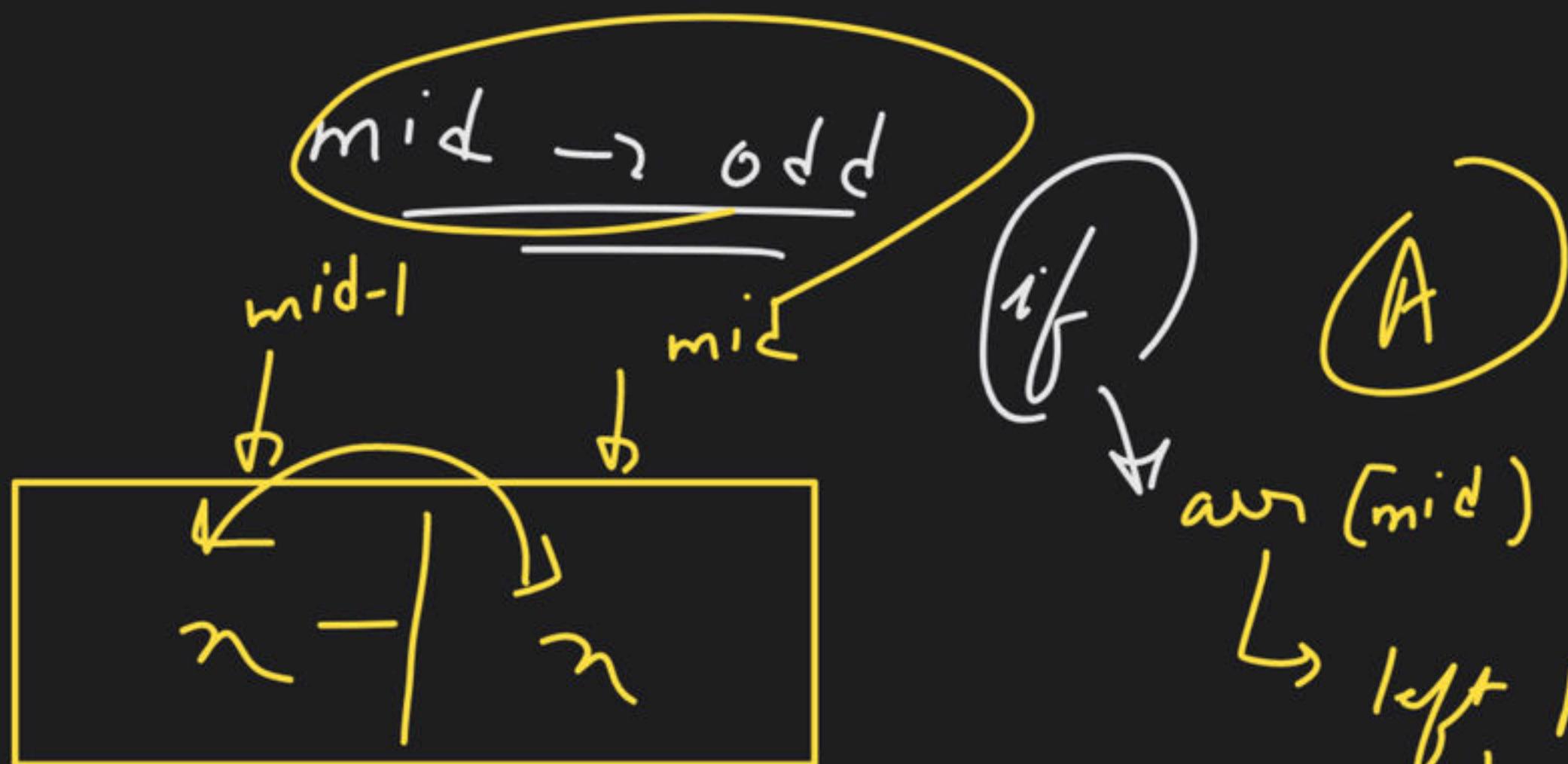
    if ( $arr[mid] > arr[mid + 1]$ )

        s = mid + 2;

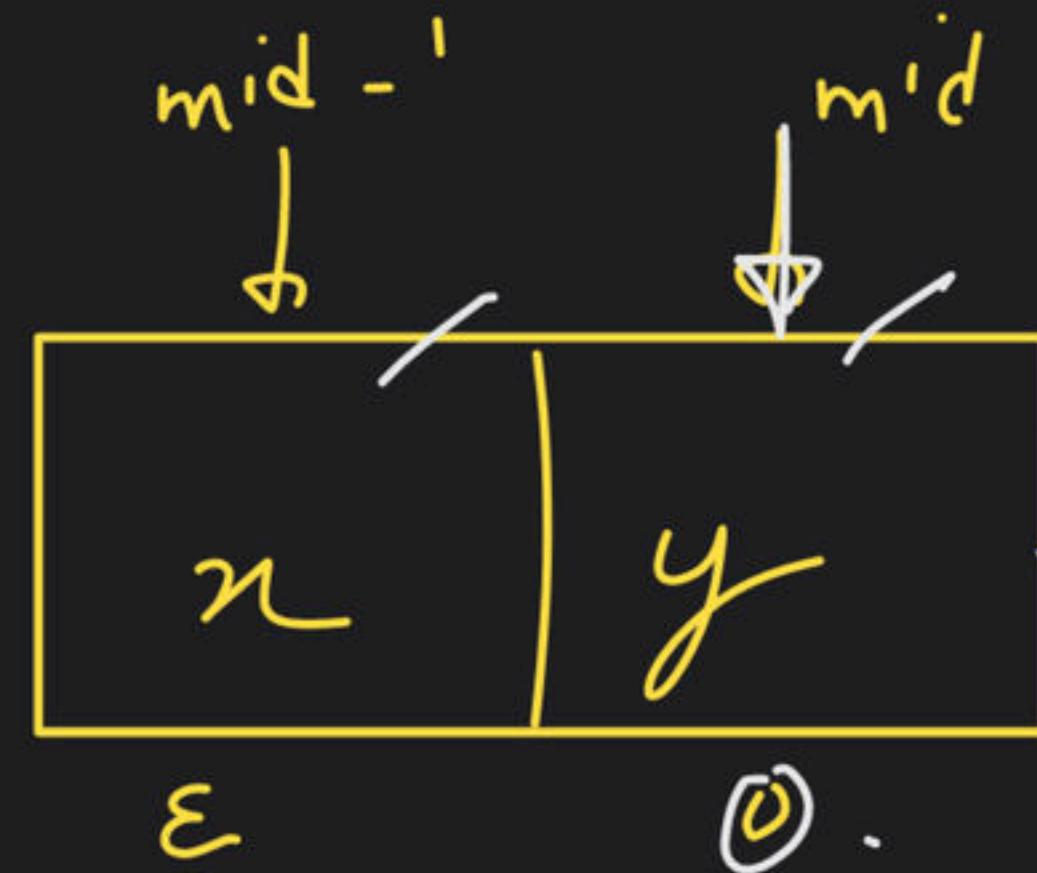
else

    e = mid;

3



E      O



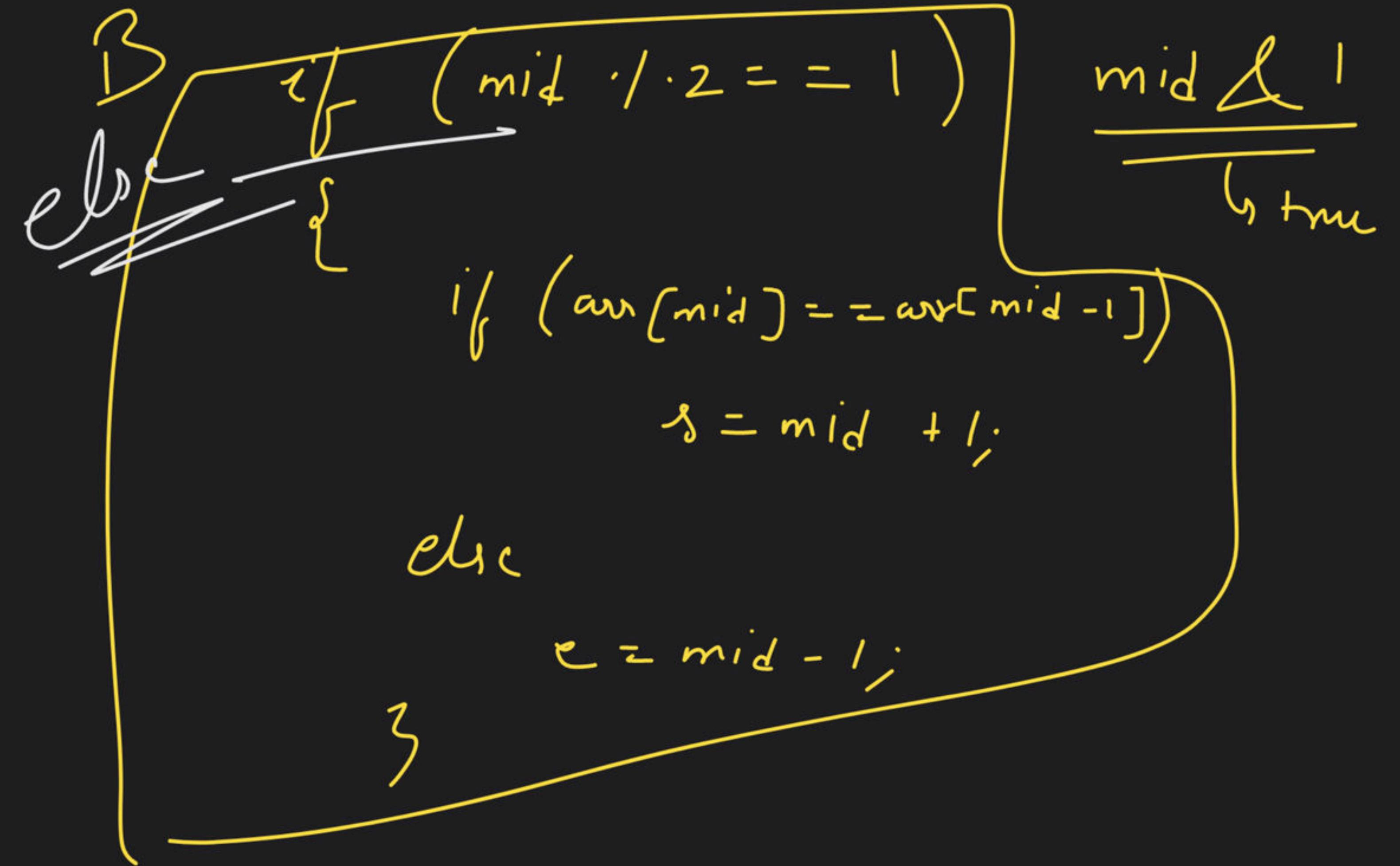
E      O.

ans      postKlade hn

ya      tol      aap      right re Klade hn

L.S  $\rightarrow$  c = mid - 1

$$j = \text{mid} + 1;$$



int s = 0;

int e = arr.size() - 1

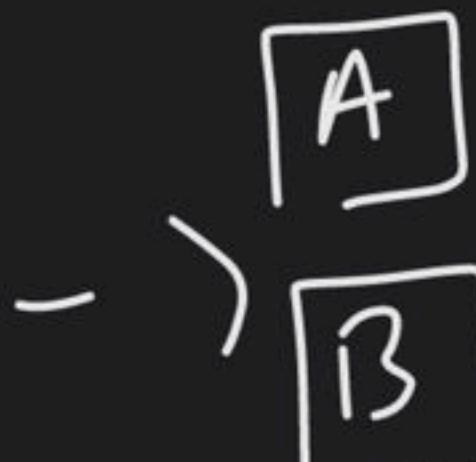
int mid = s + (e - s) / 2

while(s <= e)

{

if (s == e)

return arr[s];



mid = s + (e - s) / 2

}

return -1;

s < e



