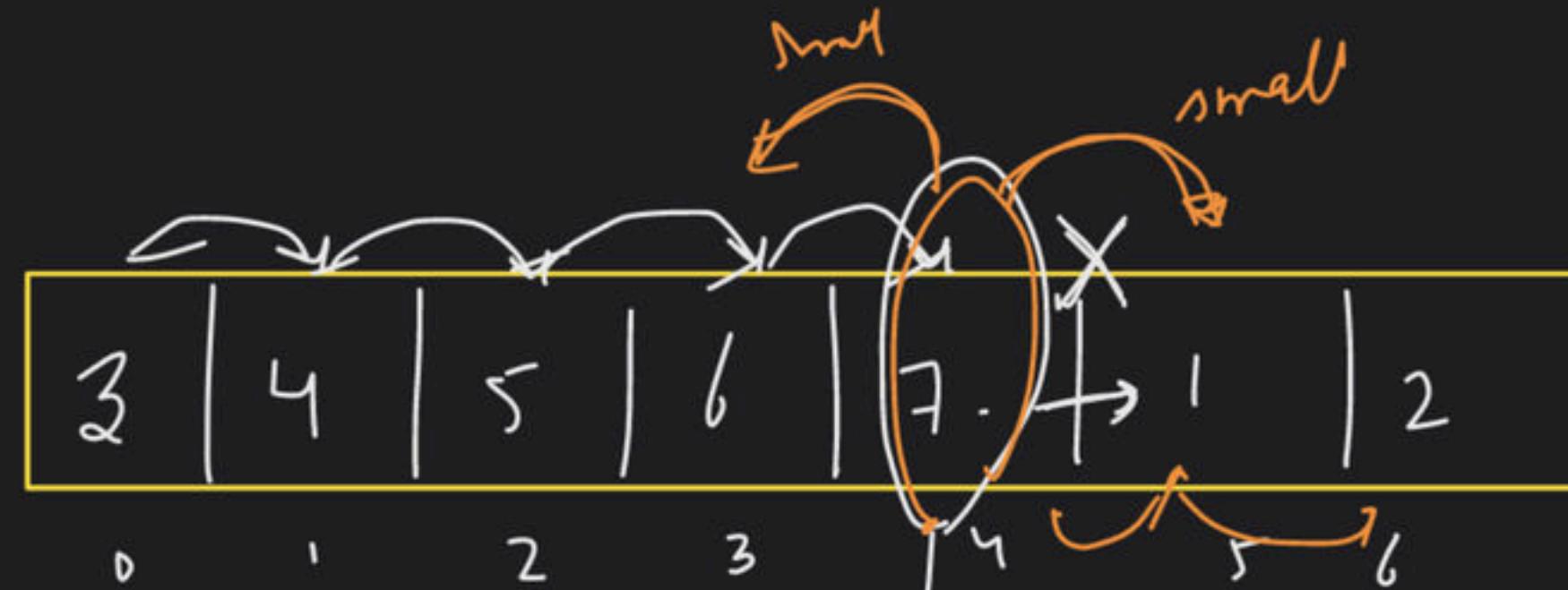


# Searching and Sorting Class - II

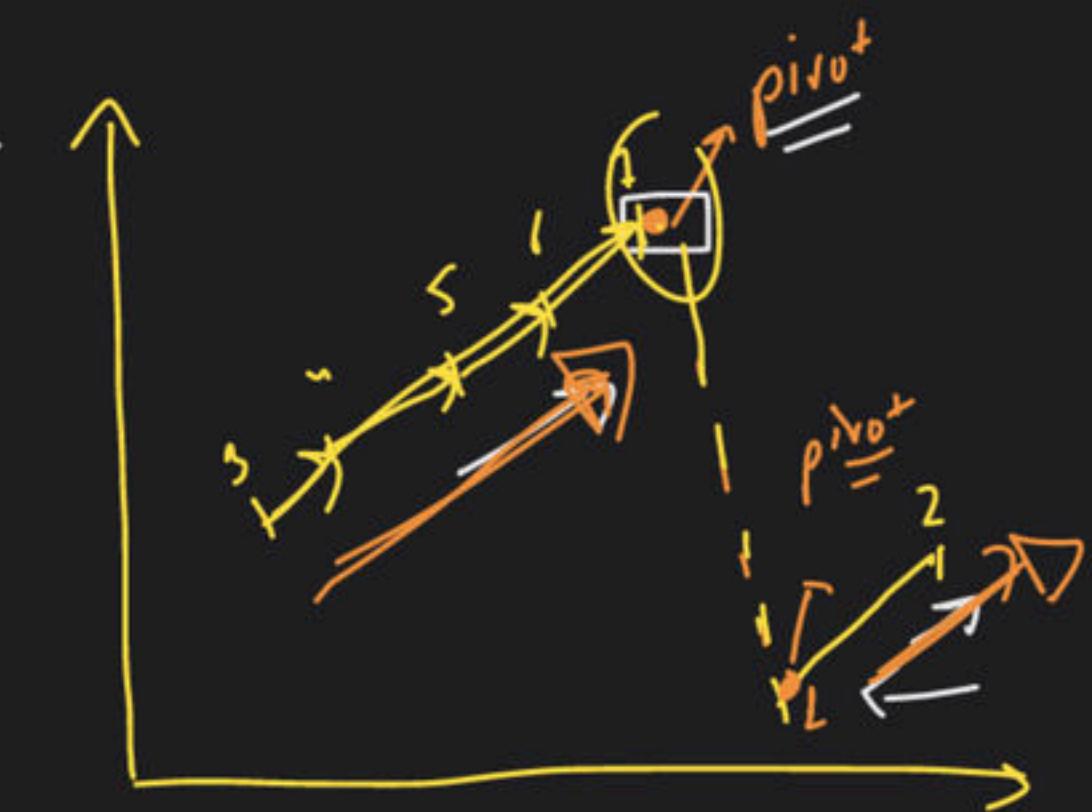
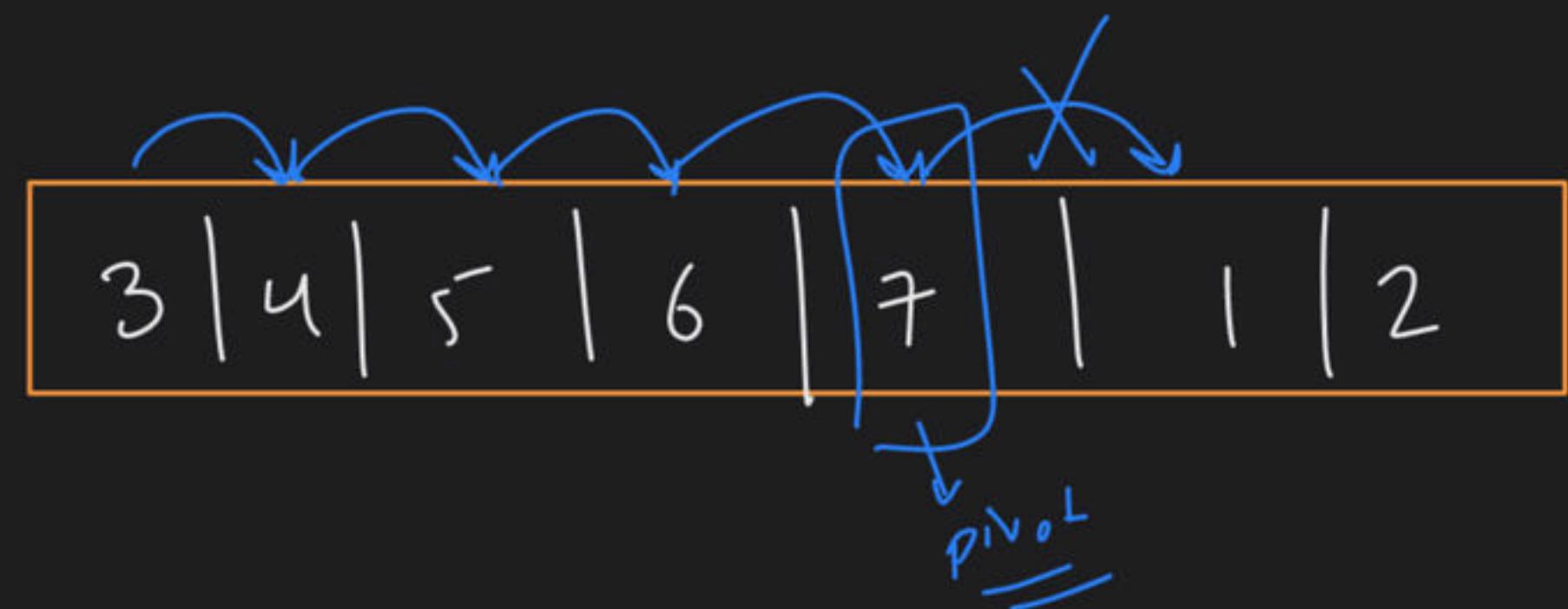
Special class

~~Binary Search~~  
T.C  $\rightarrow O(\log n)$   
first occ  
last occ  
total occ  
miss elem  
peak element

→ find pivot element



what is pivot?



Brute  
Force

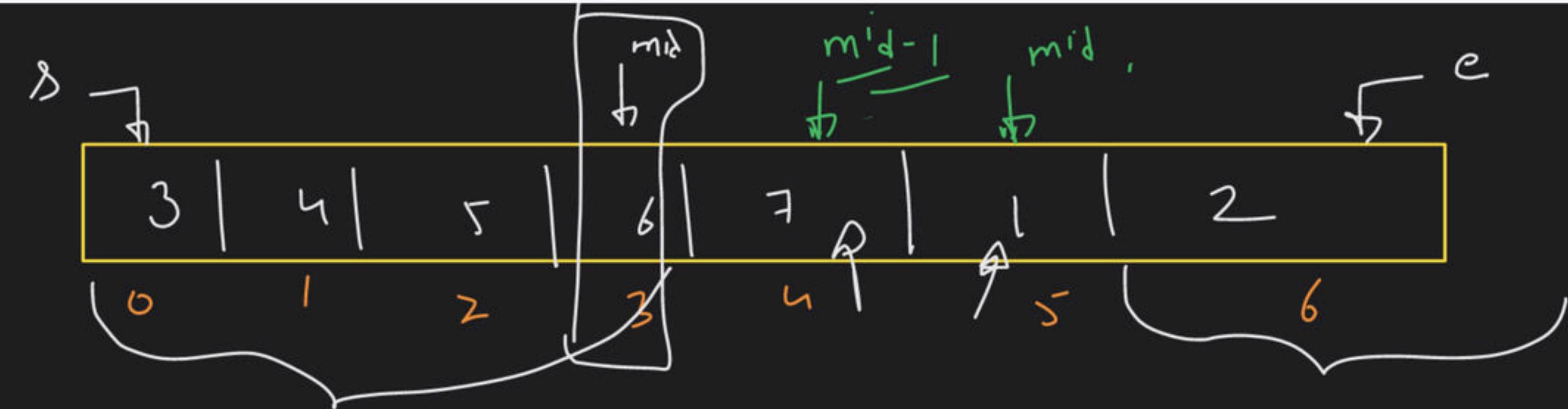
Linear  
Search

$$T \cdot C \rightarrow O(n)$$

$$\delta = 0$$

$$e = 6$$

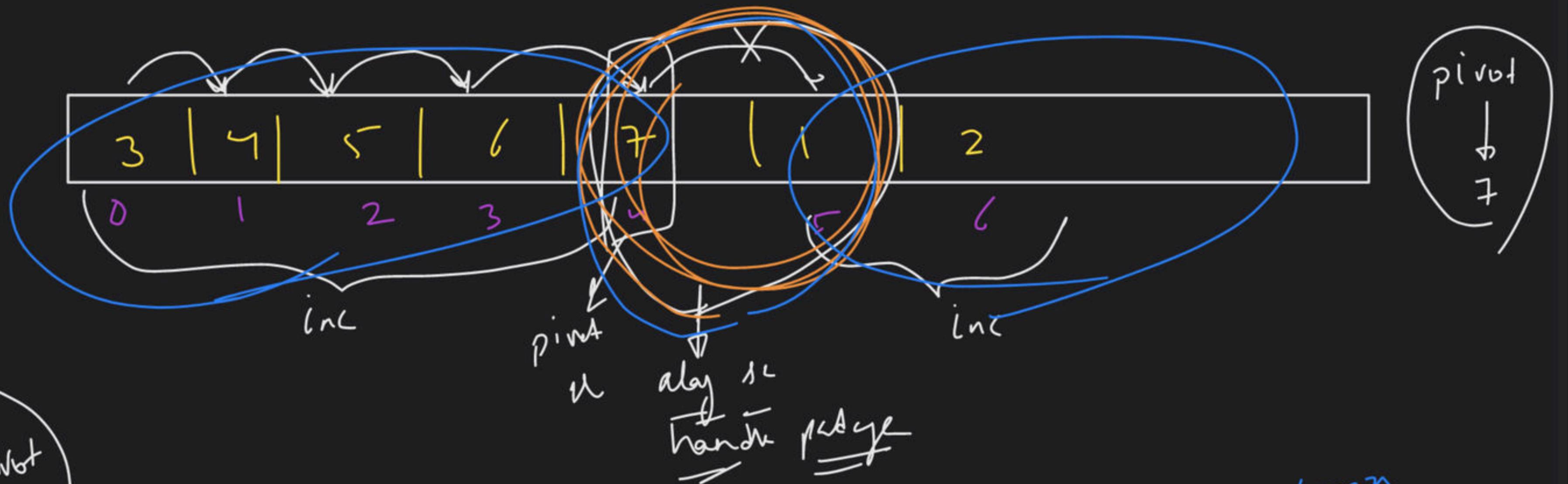
$$\text{mid} = \frac{0+6}{2} = 3$$



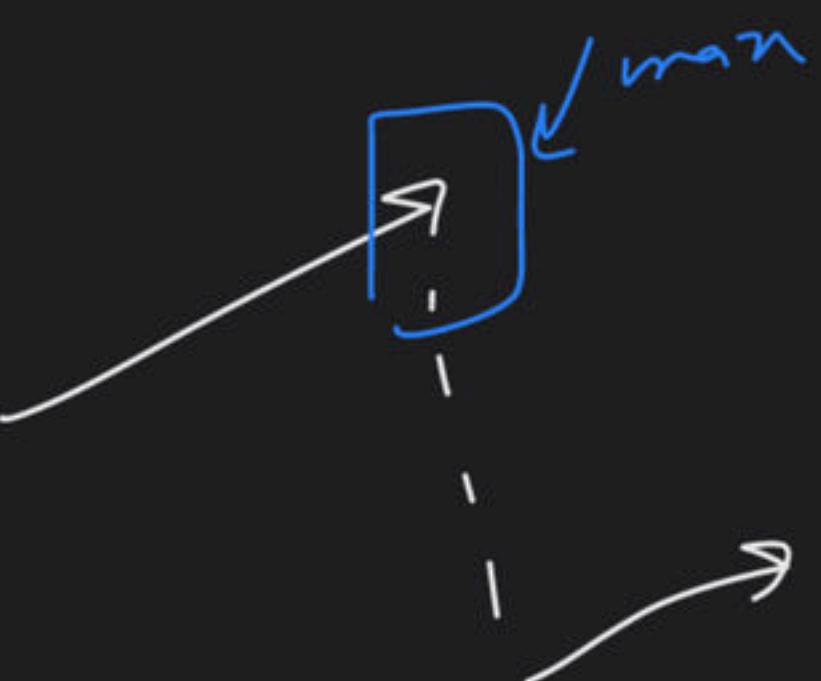
if ( $\text{arr}[\text{mid}] > \text{arr}[\text{mid}+1]$ )  
    return  $\text{mid};$

if ( $\text{arr}[\text{mid}-1] > \text{arr}[\text{mid}]$ )  
    return  $\text{mid}-1;$





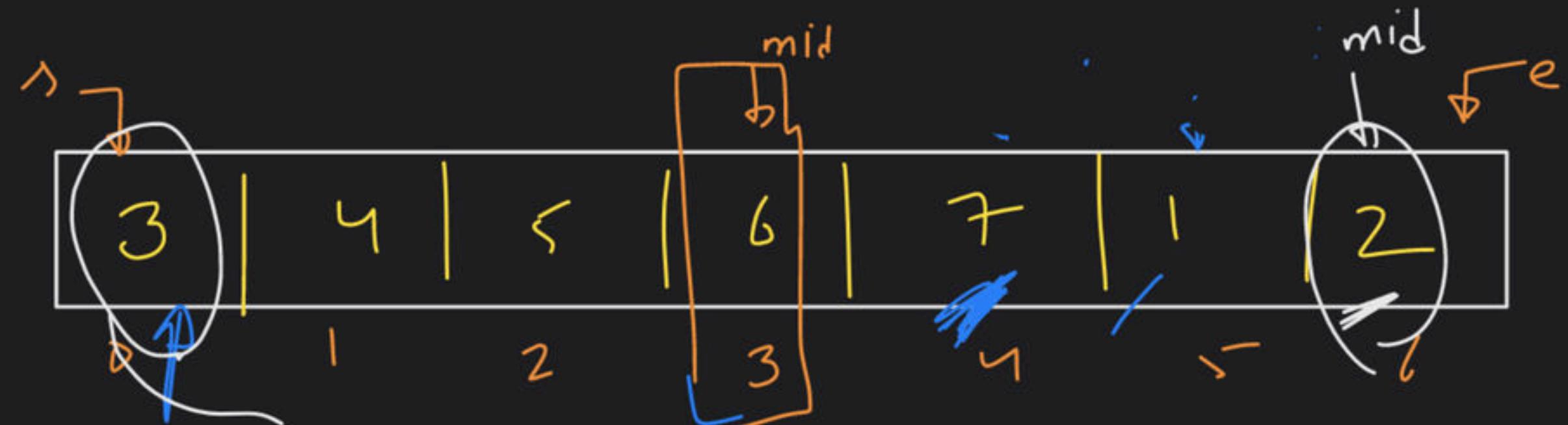
find the pivot element



$$s = 0$$

$$e = 6$$

$$\text{mid} = \frac{0+6}{2} - 3$$



if (arr[s] > arr[mid]) // left  
{  
 s = mid - 1  
}

else if (arr[r] < arr[mid]) // right

$$s = mid + 1$$

}

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

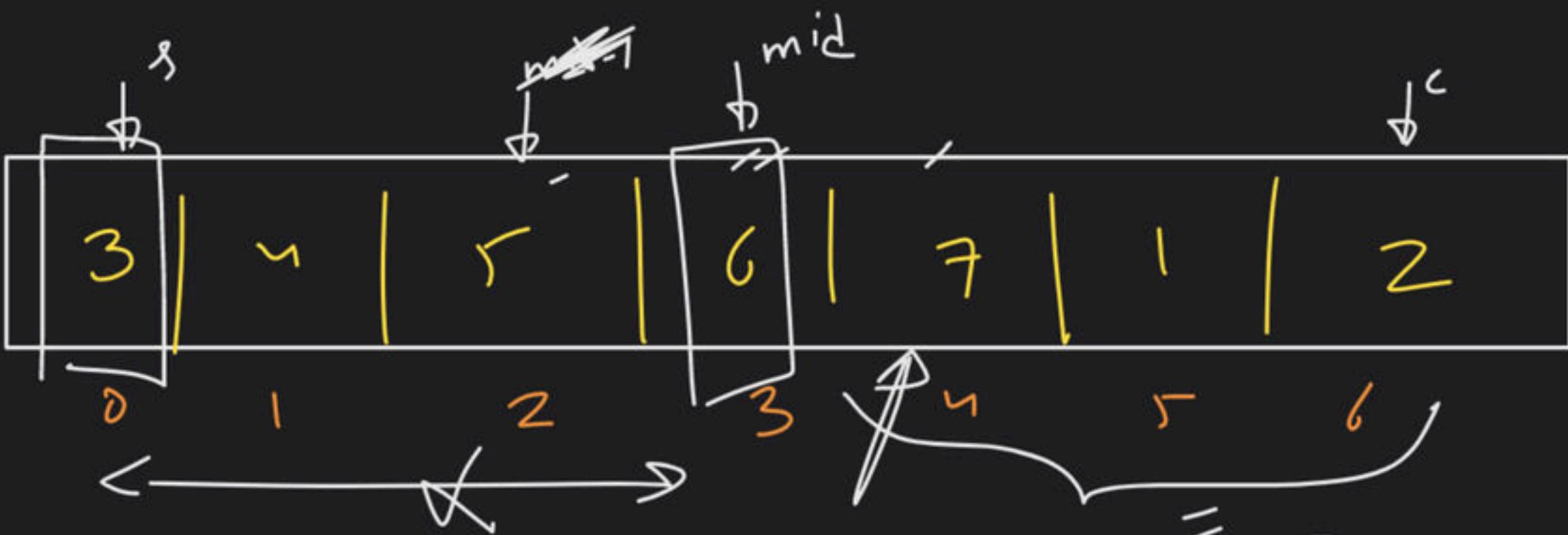
if (arr[mid-1] > arr[mid])  
 return mid - 1

else  
 56 / 1

$$d = 0$$

$$c = 6$$

$$\text{mid} \sim \frac{6+6}{2} = 3$$



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

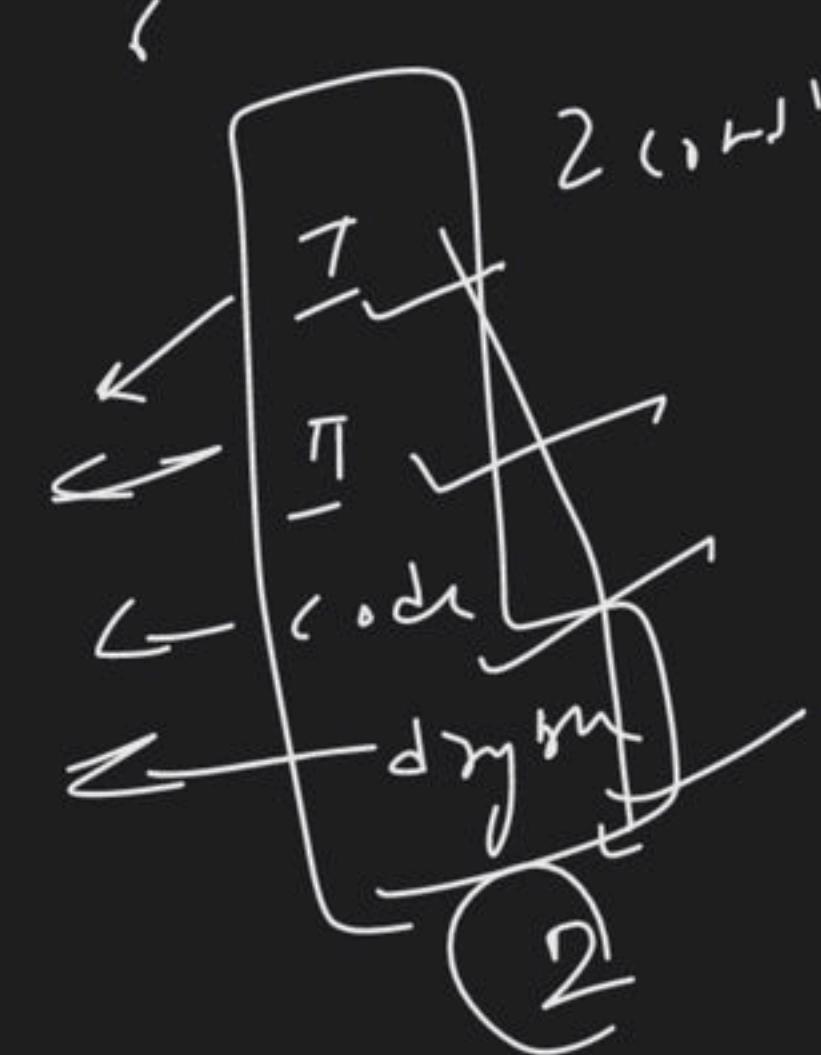
$$d = 4$$

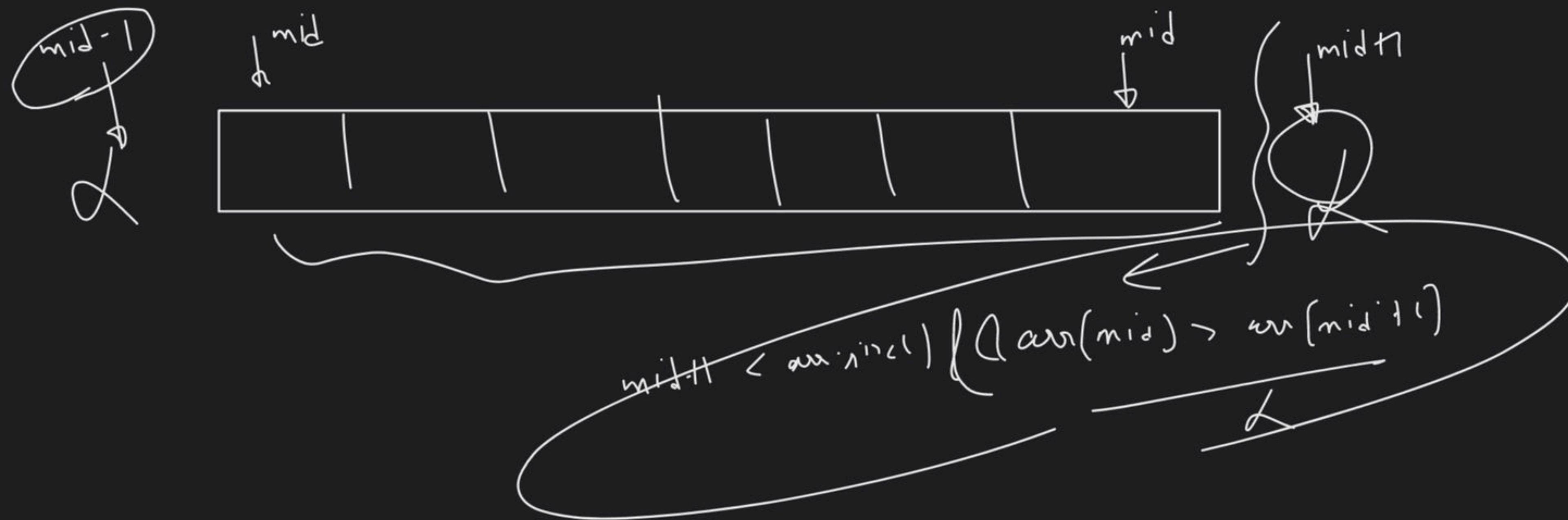
$$c = 6$$

$$\text{mid} + 4 + 1 \rightarrow 5$$

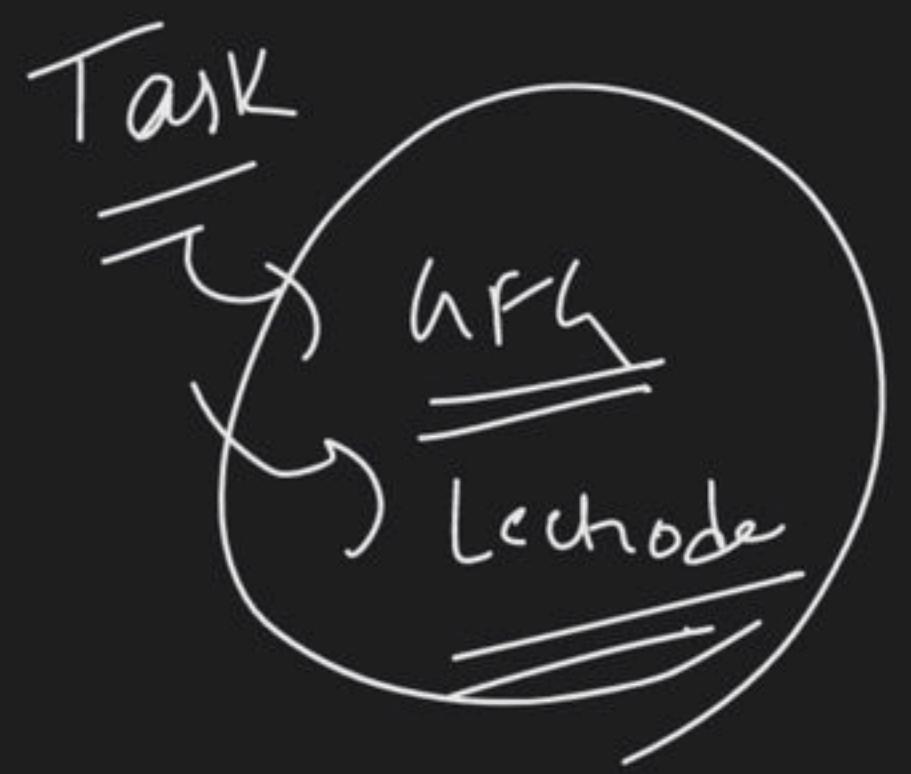


return





$\text{mid}-1 \geq 0 \& \& \text{arr}[\text{mid}-1] > \text{arr}[\text{mid}]$



1 min

0		1		2		3		4		5		6
---	--	---	--	---	--	---	--	---	--	---	--	---

swap

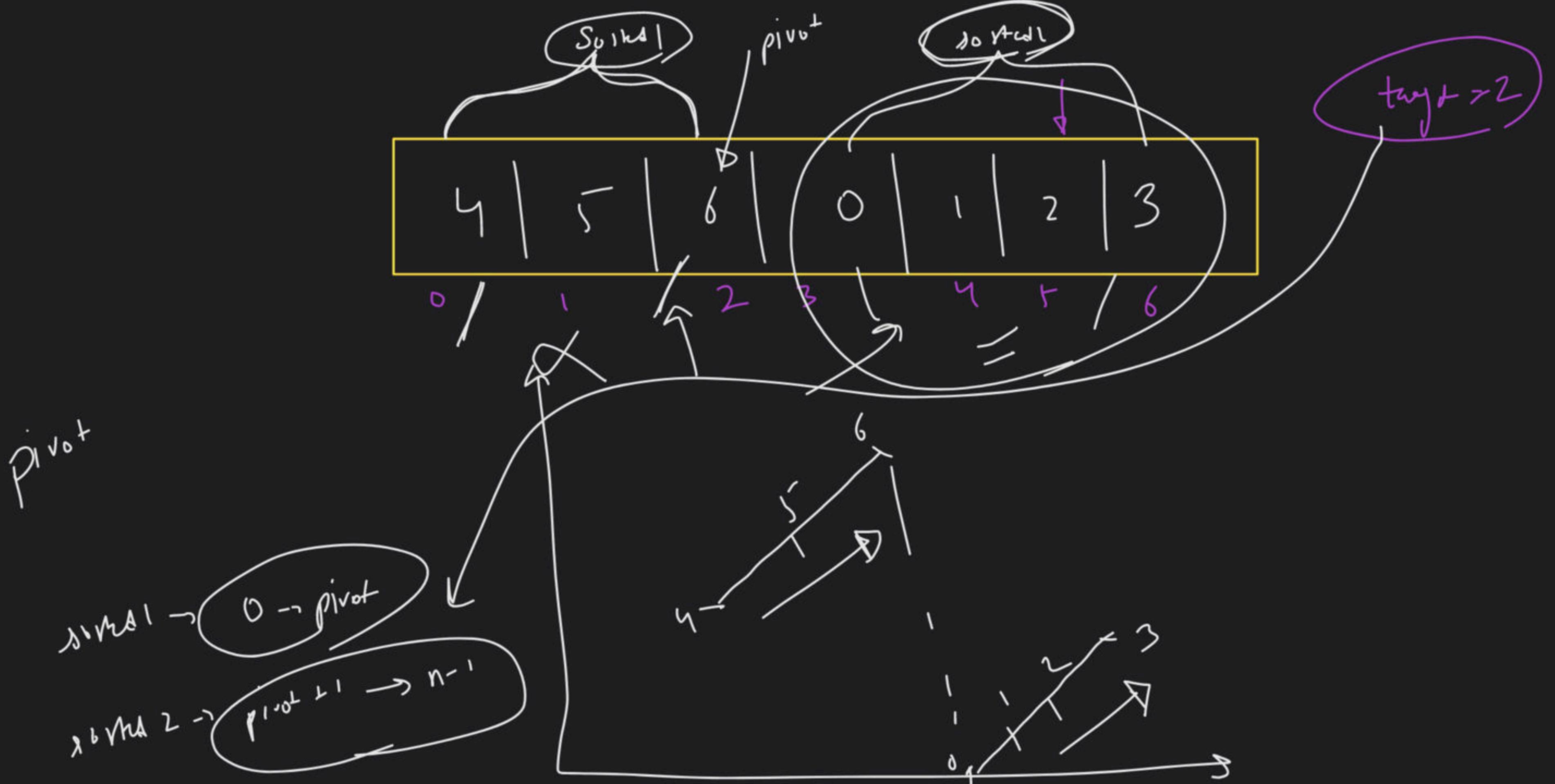
6		0		1		2		3		4		5	
---	--	---	--	---	--	---	--	---	--	---	--	---	--

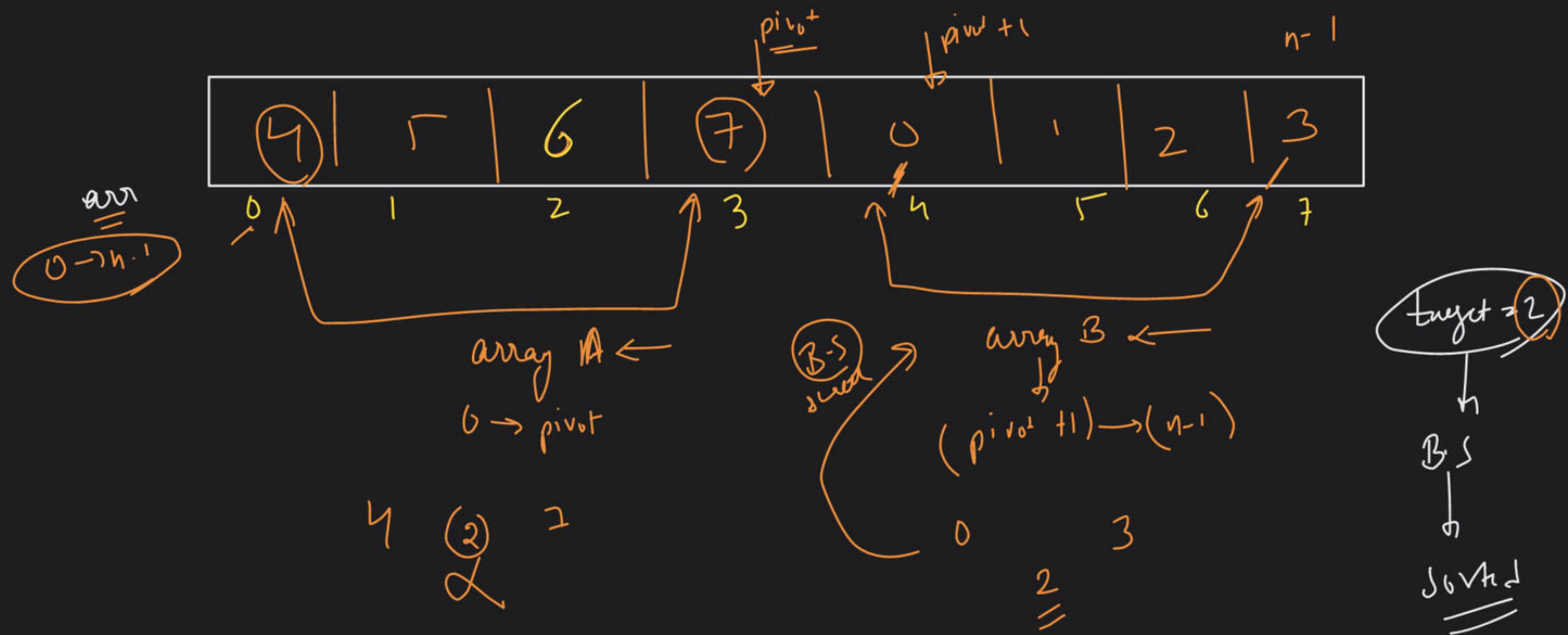
5		6		0		1		2		3		4
---	--	---	--	---	--	---	--	---	--	---	--	---

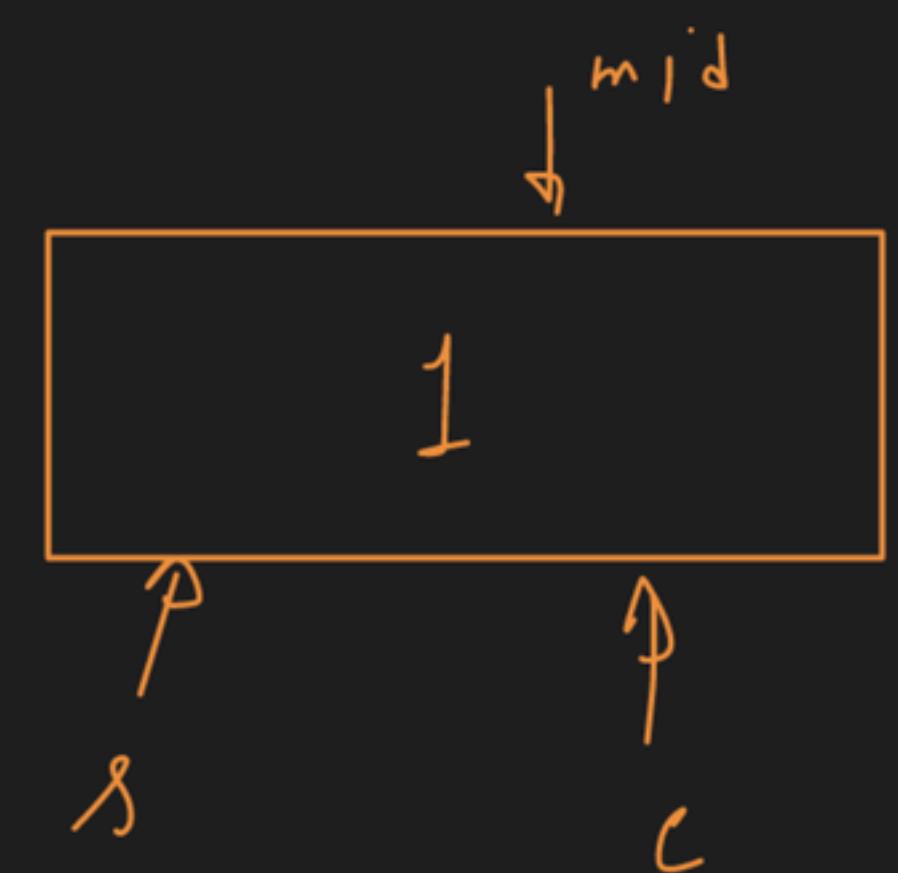
arr

4		5		6		0		1		2		3
---	--	---	--	---	--	---	--	---	--	---	--	---

rotate



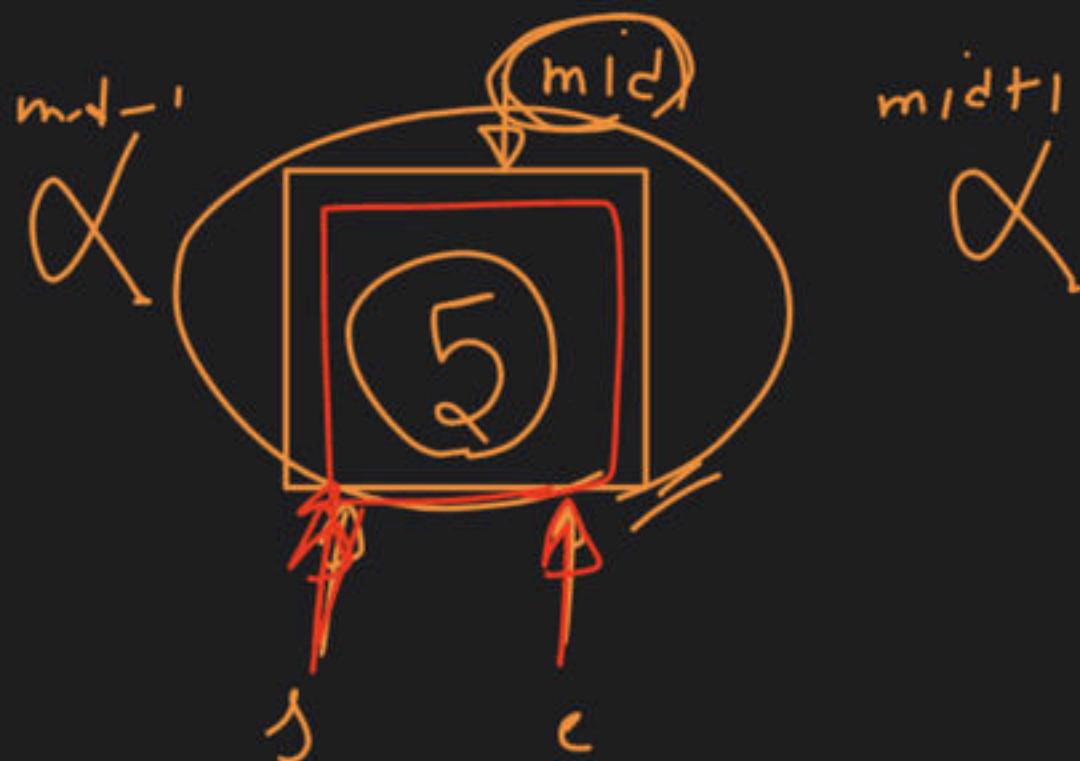




$$s = 0$$

$$c = 0$$

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



if (arr[s] >= arr[mid])  
    / High search  
    c = mid - 1

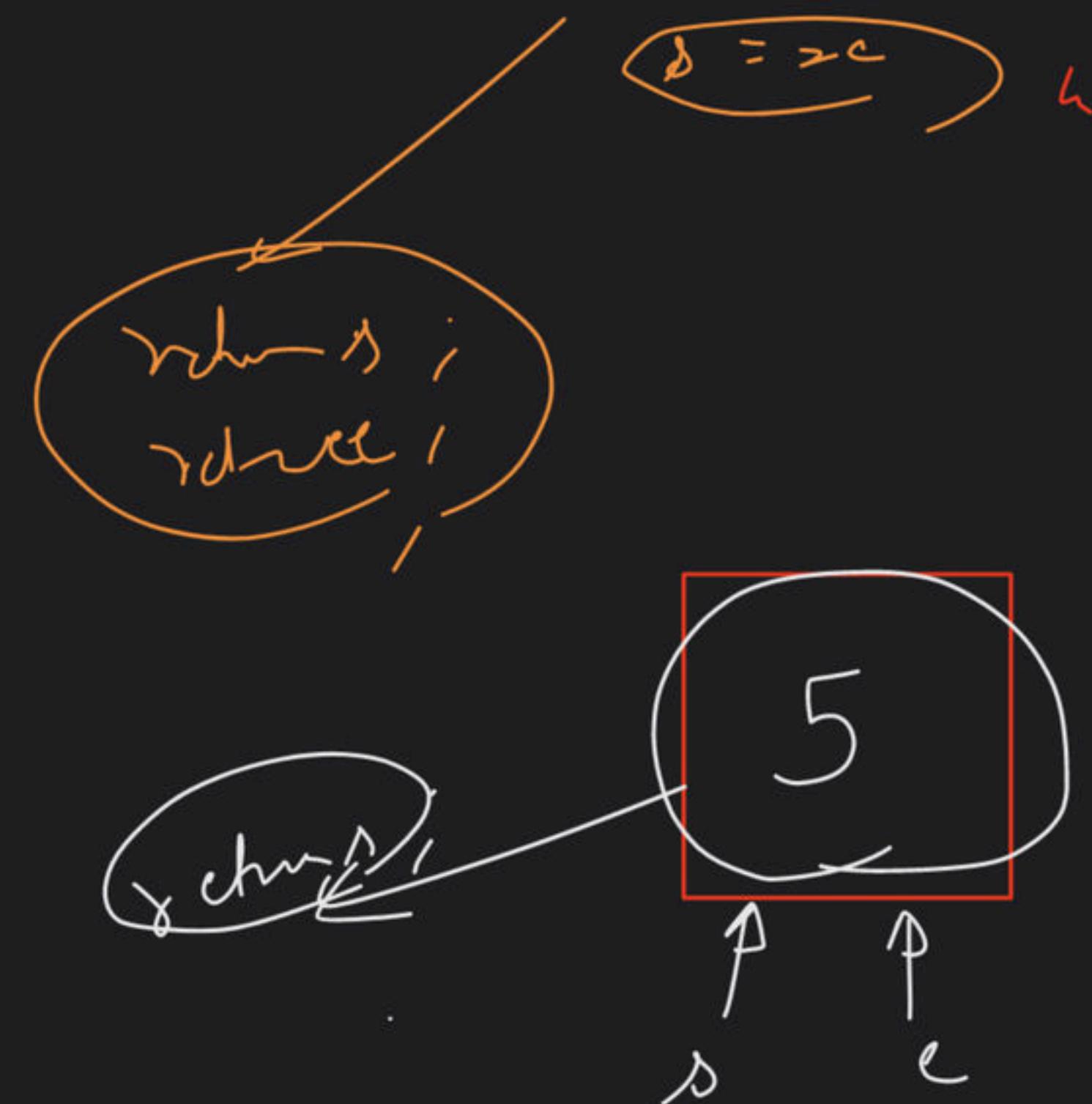
else

s = mid + 1

mid + 1

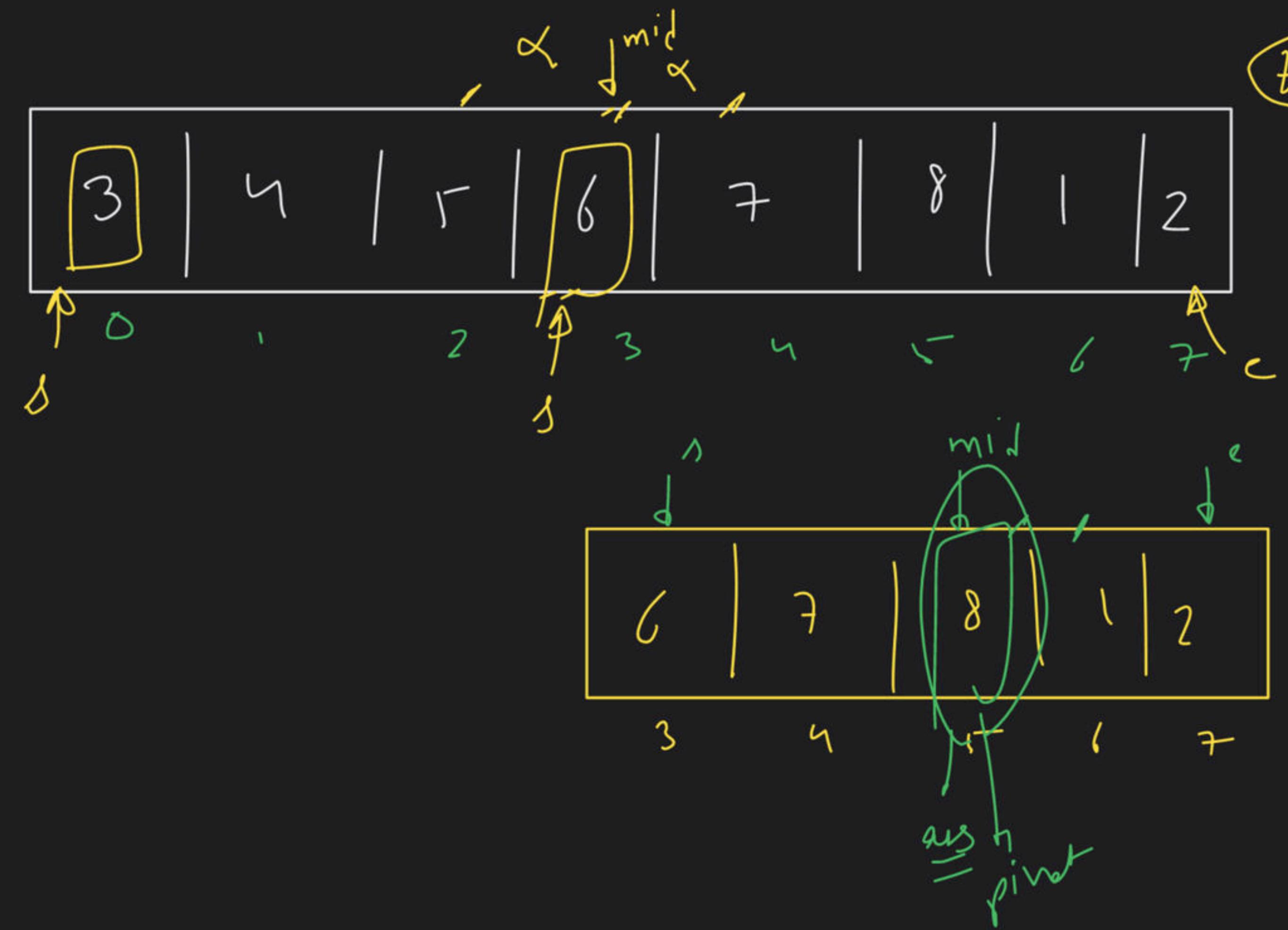
while (s < c)

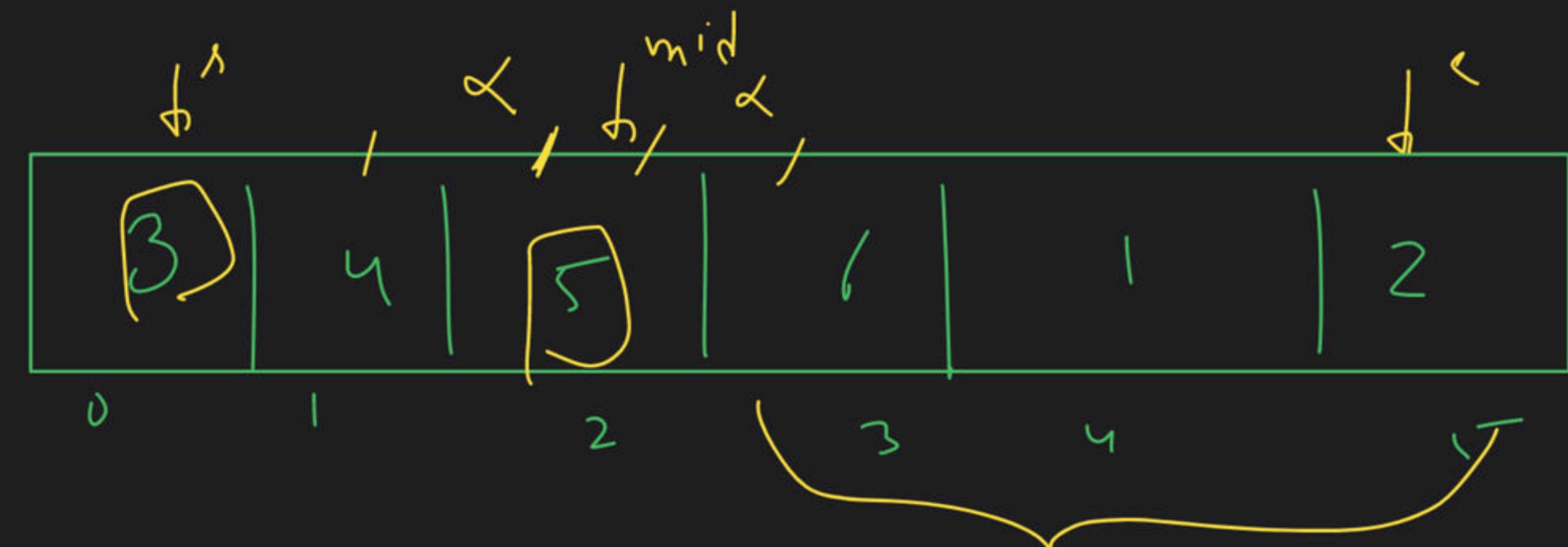
=



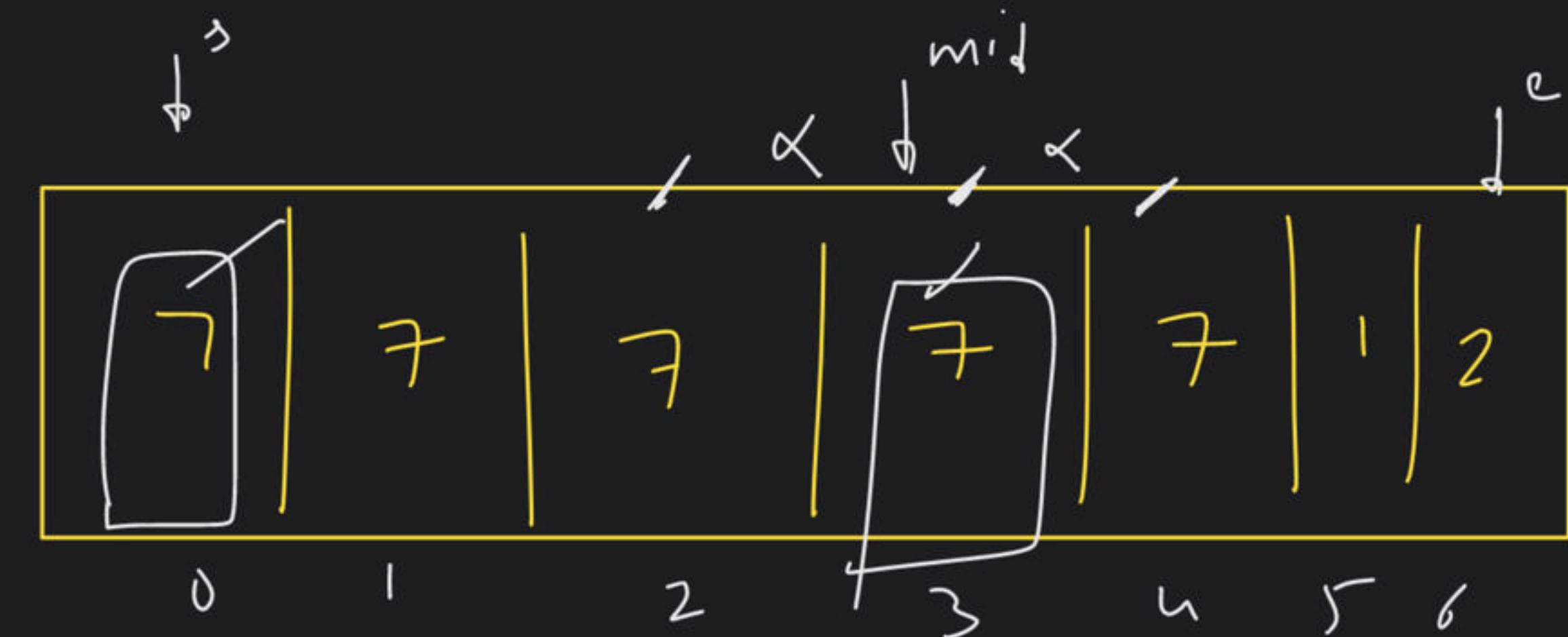
s chnge

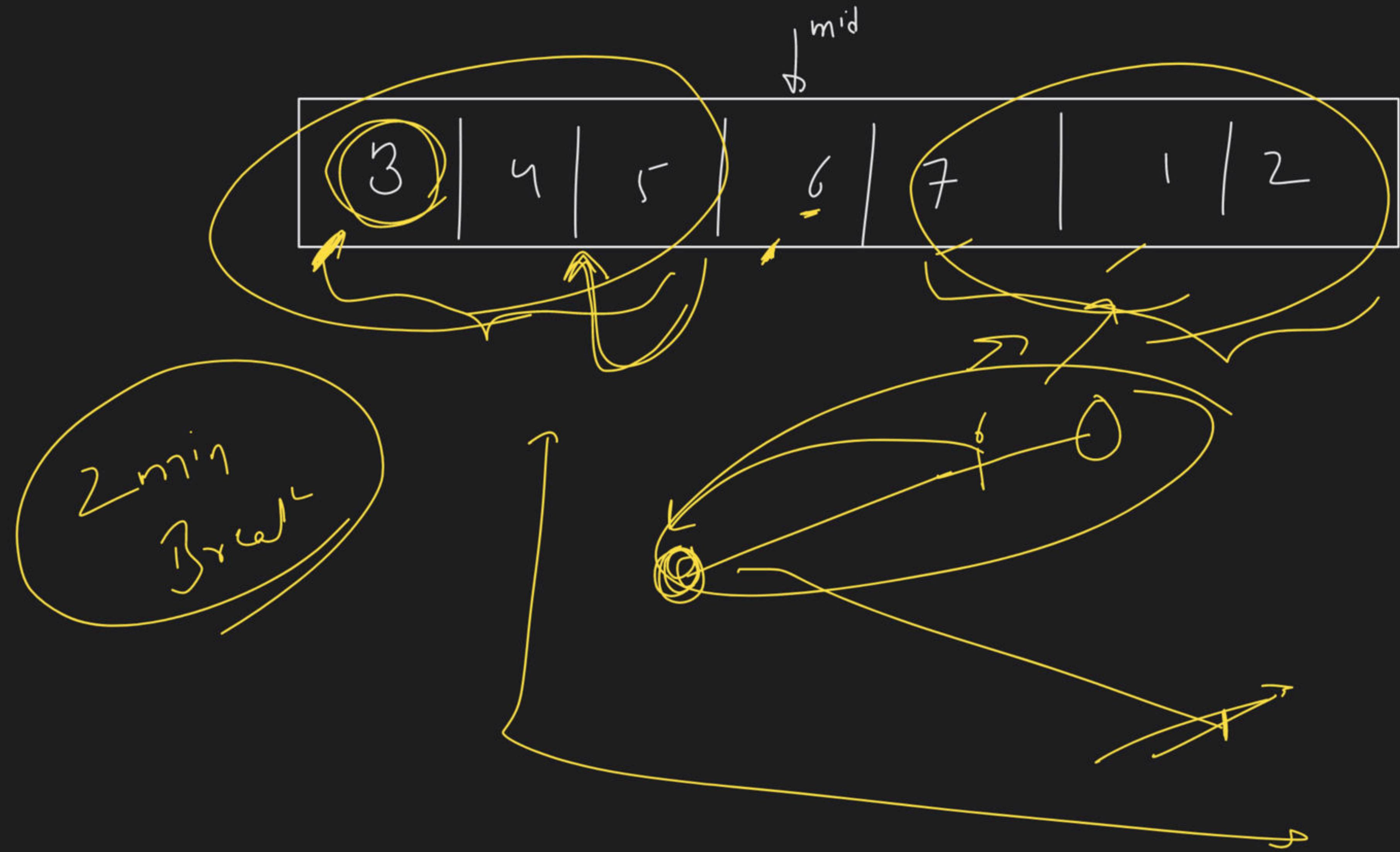
return s;  
done;



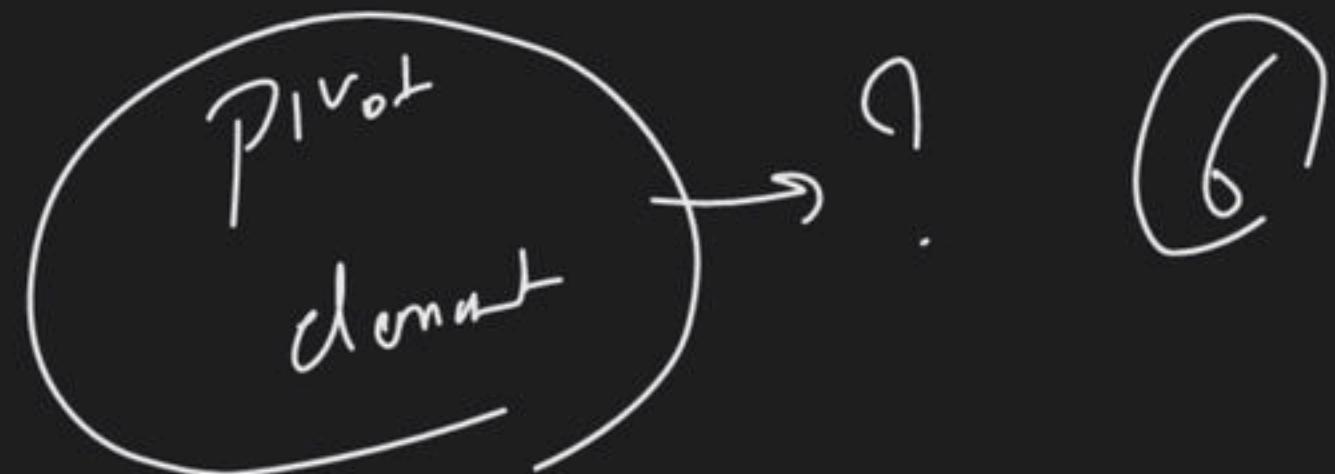
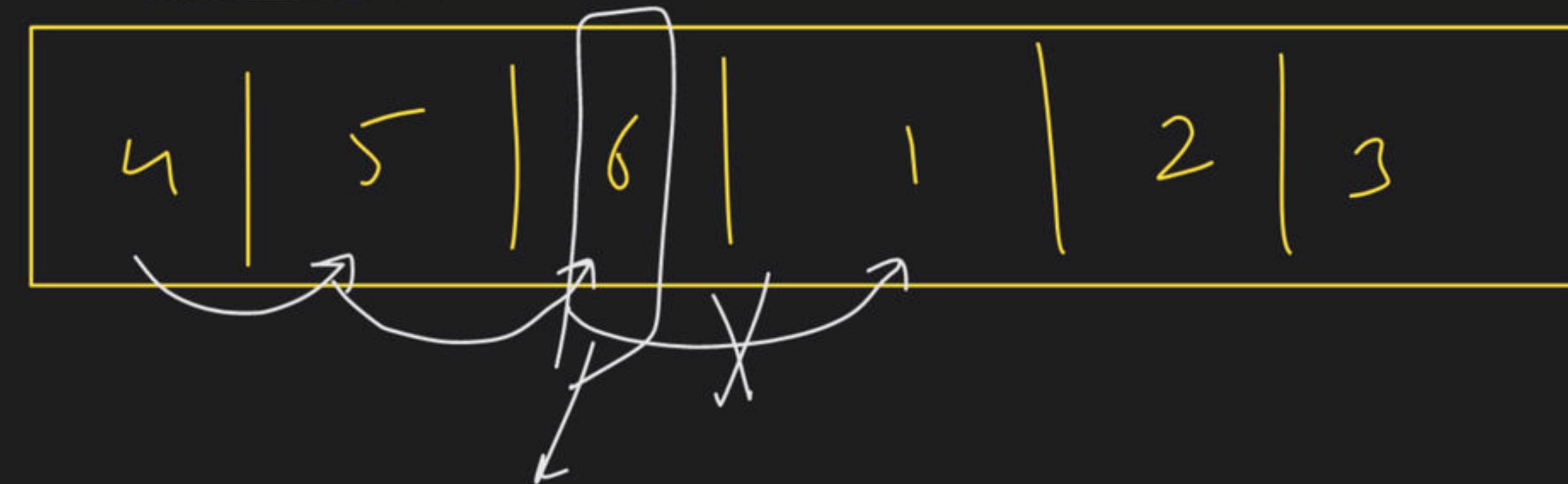
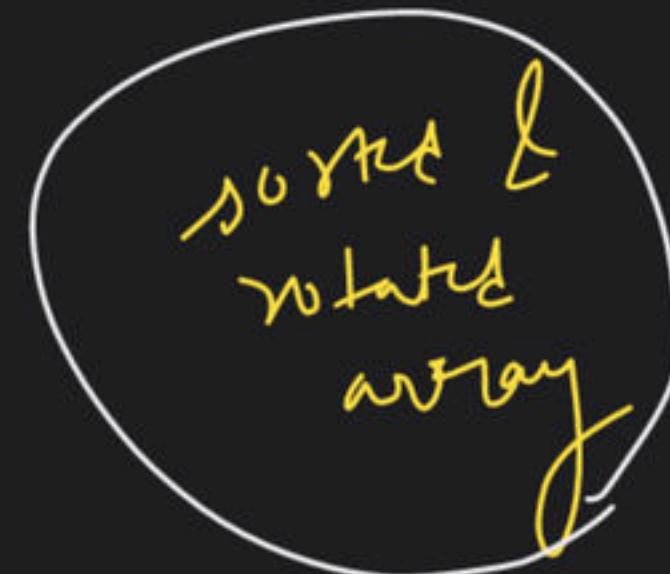


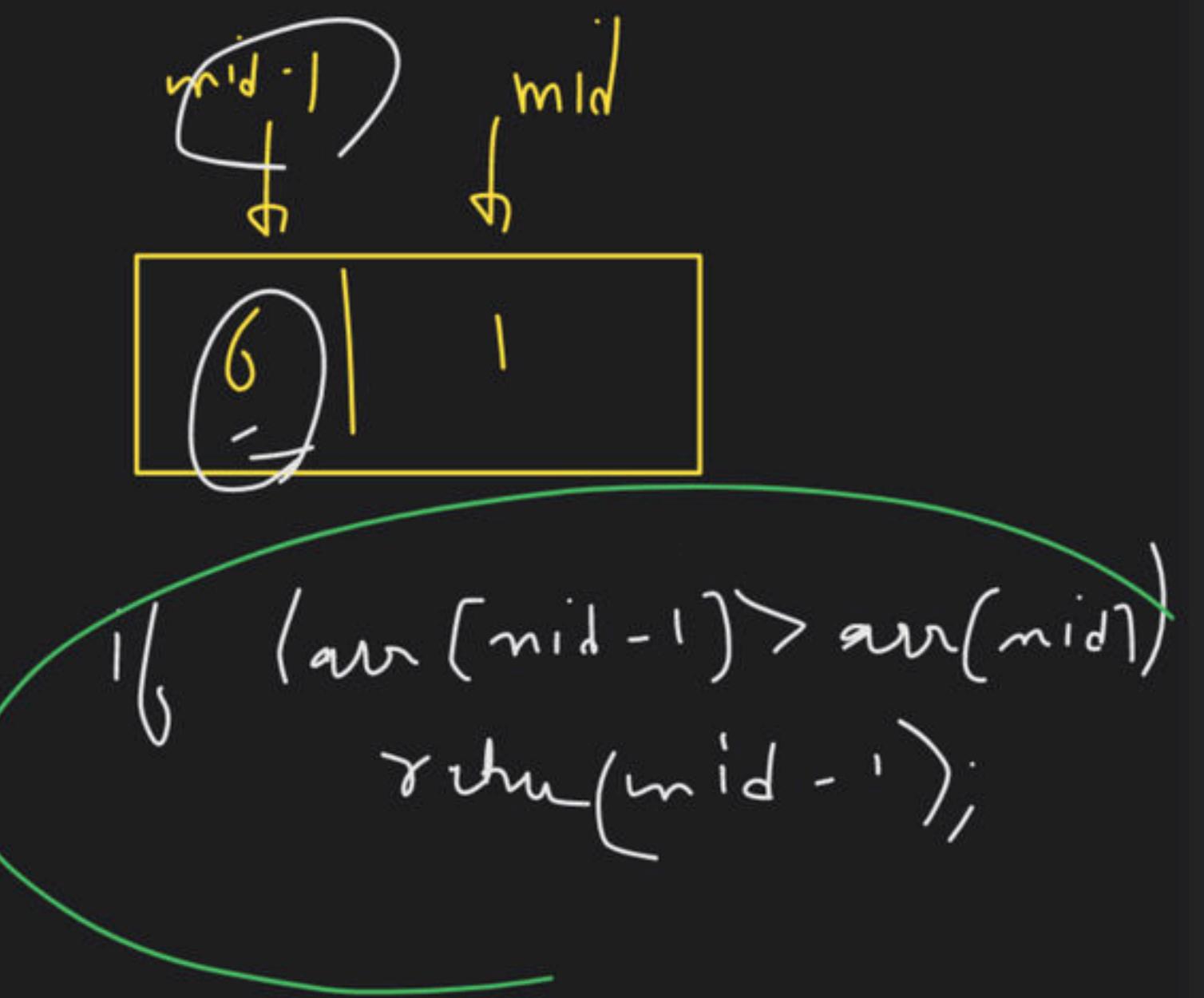
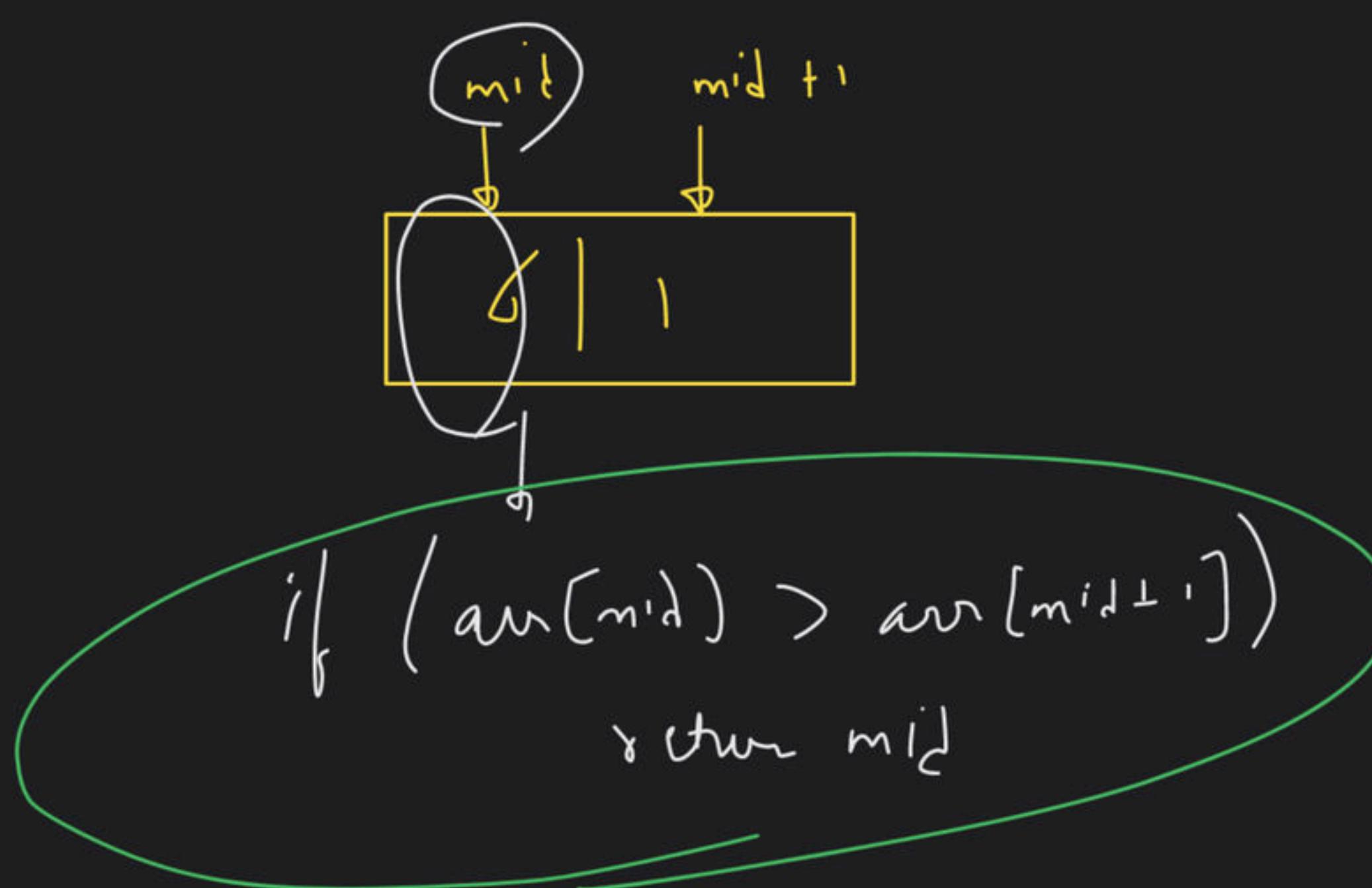
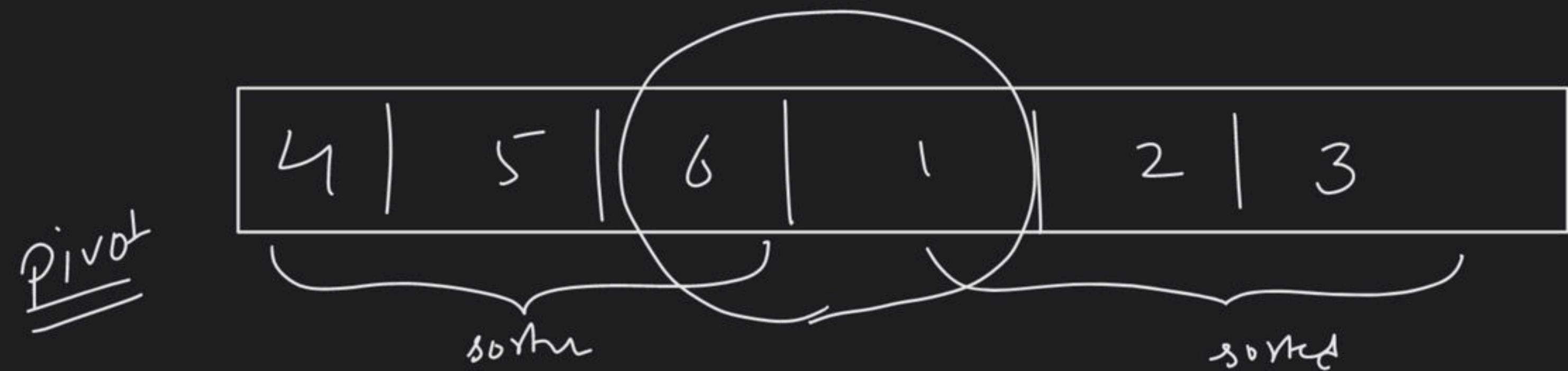
$\Delta = 0$   
 $c = 6$   
 $mid = \frac{0+1}{2} = 0.5$





86 ms



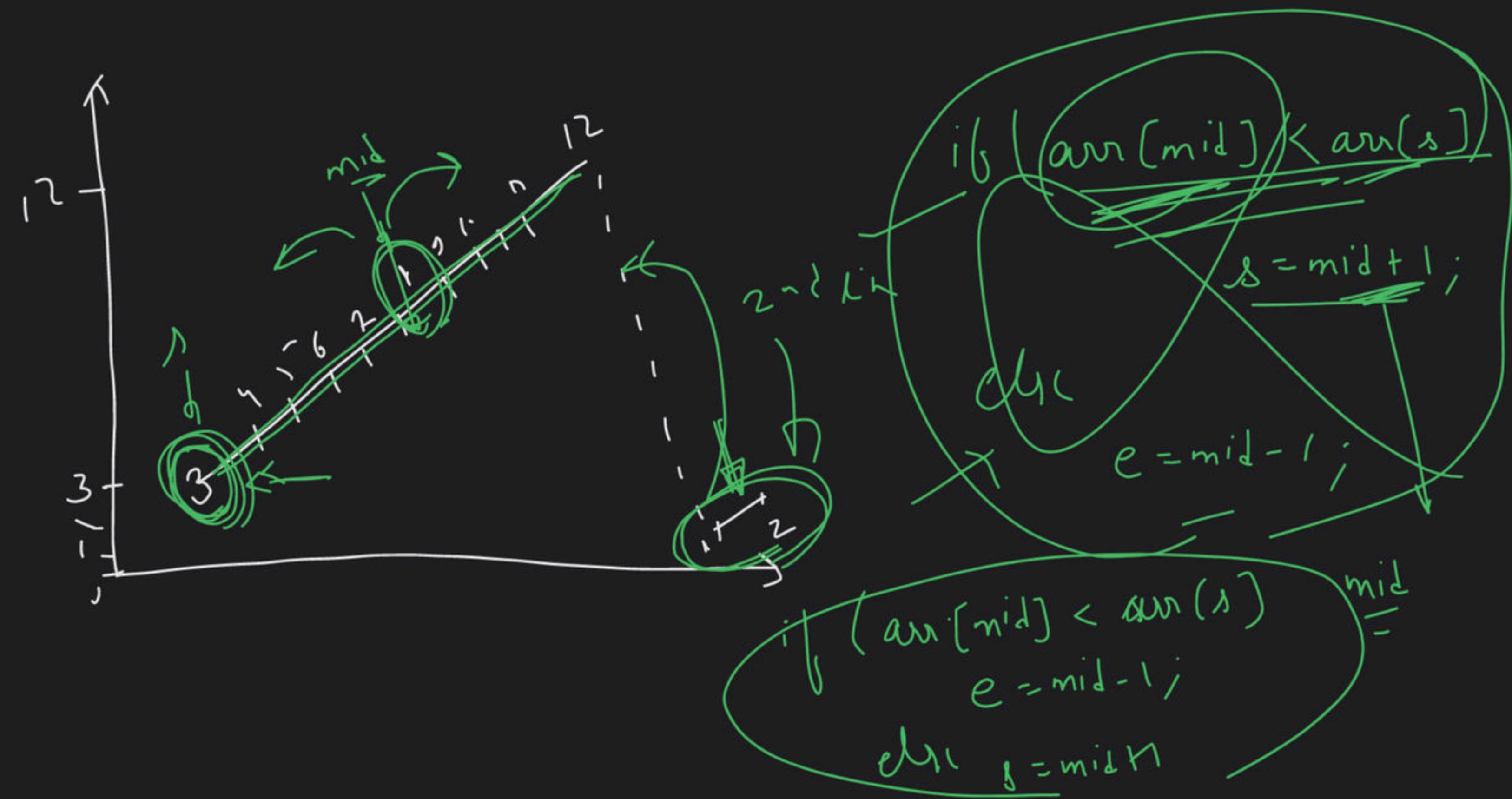
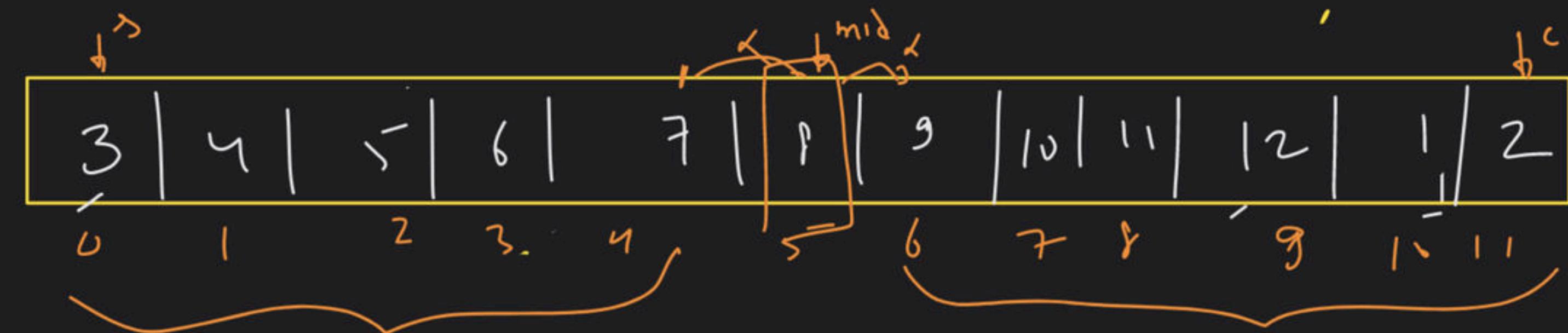


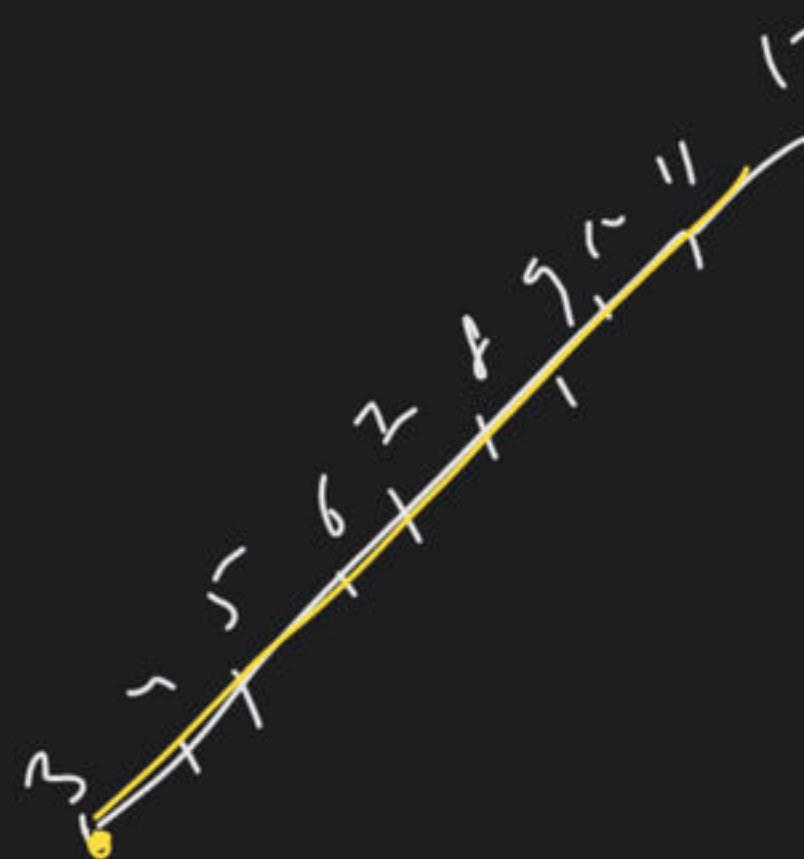
$$\delta = 0$$

$$c = 11$$

arr

$$\text{mid} = \frac{0+11}{2} = 5$$





12

11

10

9

8

7

6

5

4

3

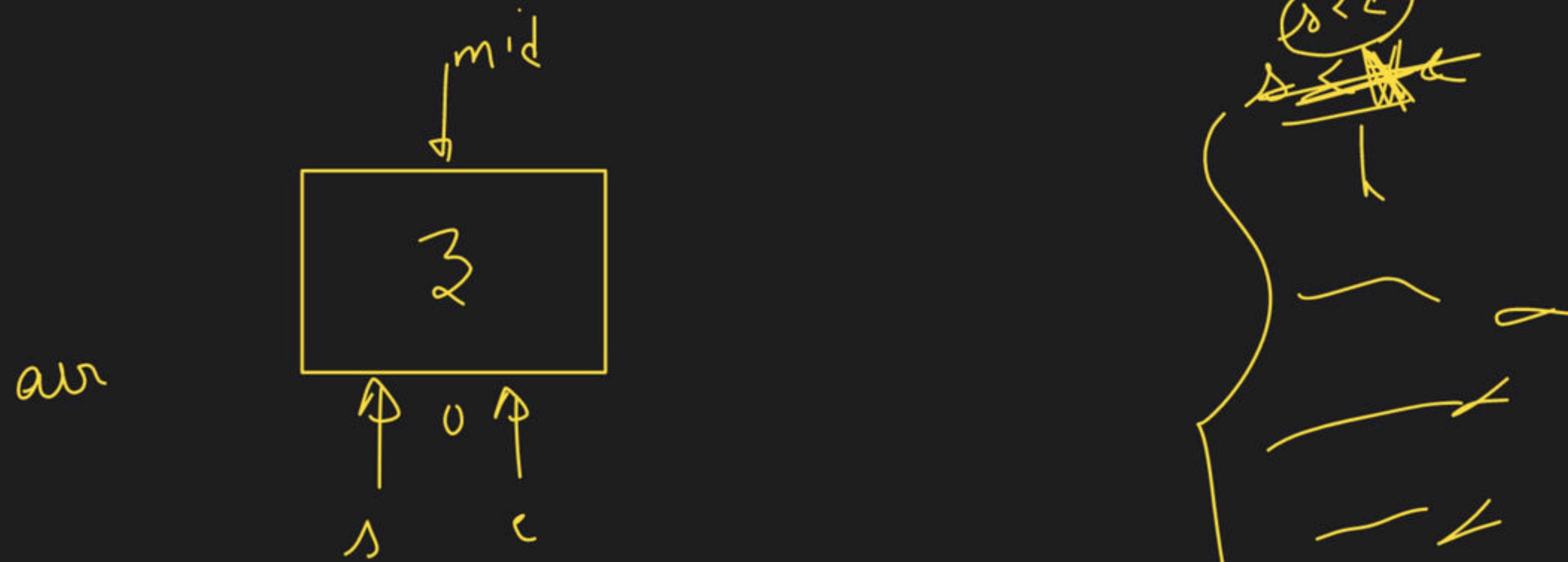
2

1

2

1

d<sub>hi</sub> $\text{if } (\underline{arr}[\text{mid}] < \underline{arr}[\text{r}])$  $e = \text{mid} - 1;$  $\delta = \text{mid} + 1;$



$s = mid + 1$

$e = mid - 1$

$1 < c$

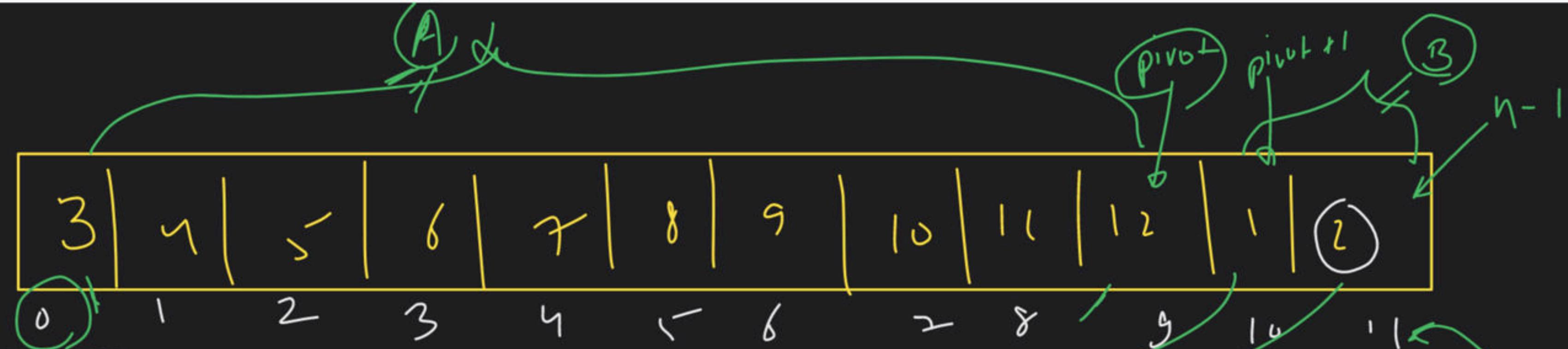
~~$s = mid$~~

$= mid$

return  $s$

else  $c$

7	6	5	4	3	2	1
---	---	---	---	---	---	---



rotates k times  
around pivot

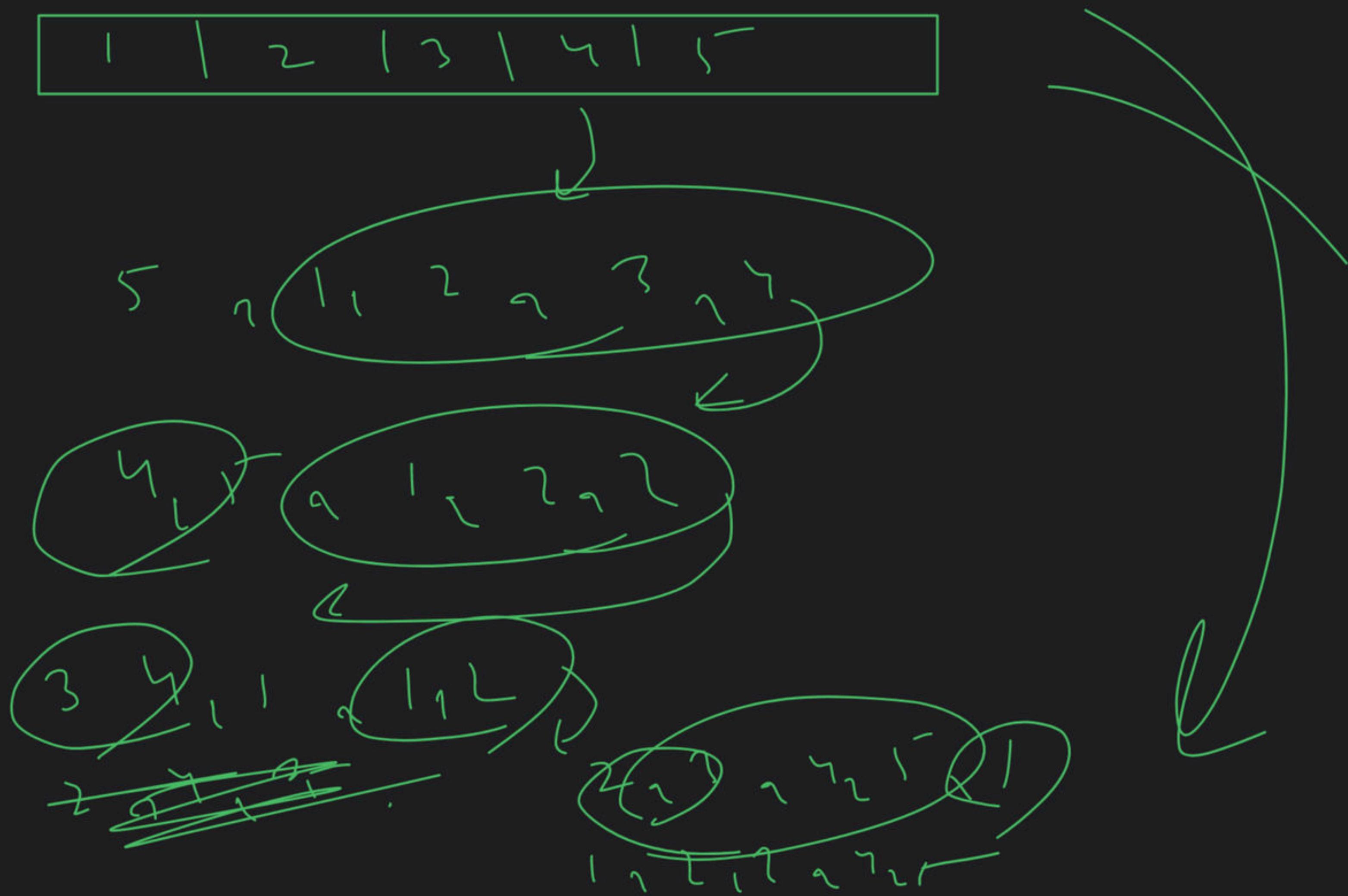
$target = 2$

1

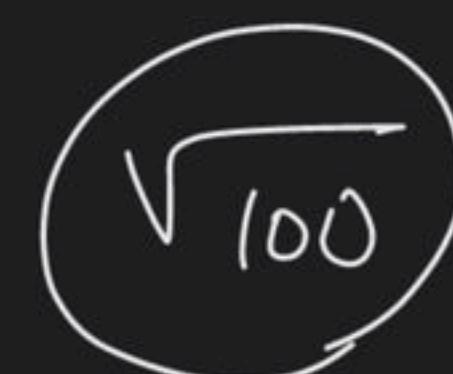
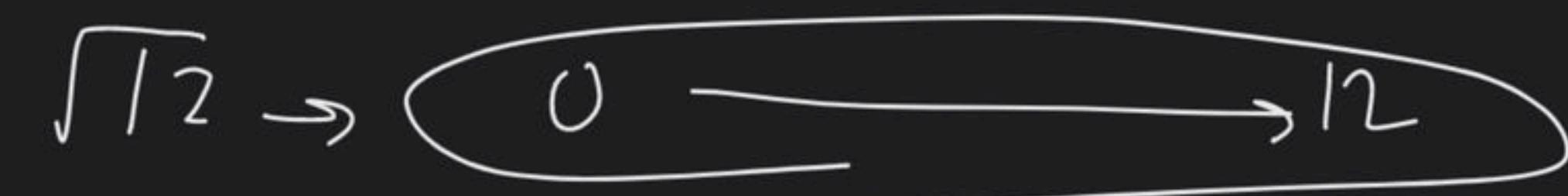
$\Delta \rightarrow 0 \rightarrow \text{pivot}$   
 $\beta \rightarrow \text{pivot} + 1 \rightarrow n - 1$

$$\log n + \log 2 = \log n$$

1		2		3		4		5
---	--	---	--	---	--	---	--	---

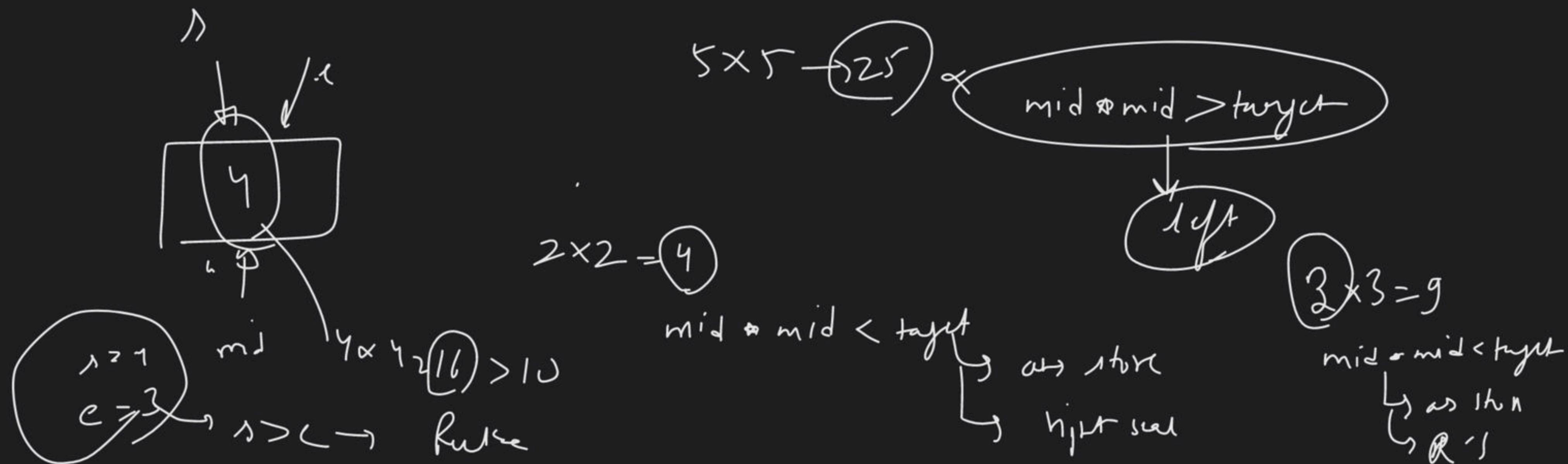
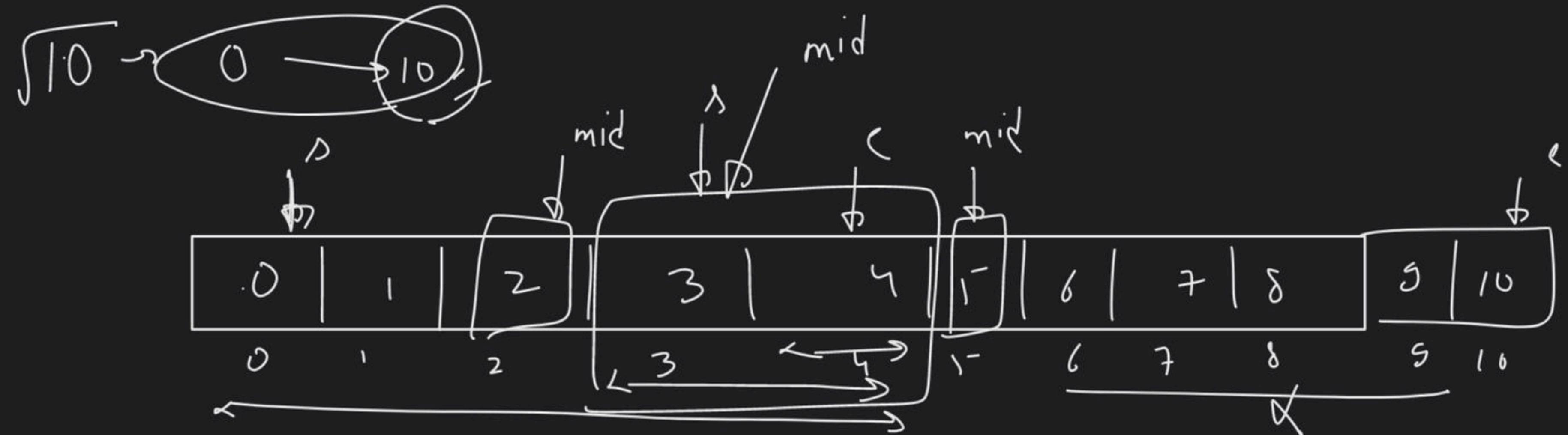


→ Square Root of a number using Binary Search



search  
space

Pivot  
Search in  
Rotated sorted  
array



if ( $\text{mid} < \text{target}$ )

↳ L.S

if ( $\text{mid} > \text{target}$ )

↳ R.S  
↳ max stone

if ( $\text{mid} & \text{mid} = \text{target}$ )

return mid

n = 25

j = 0

mid = 12

e = 25

$$12 * 12 = 144 > 25$$

↳ left son

c = mid - 1

c = 11

$$\sqrt{25} = 5$$

d = 0

mid = 5

$$5 * 5 = 25 = \text{target}$$

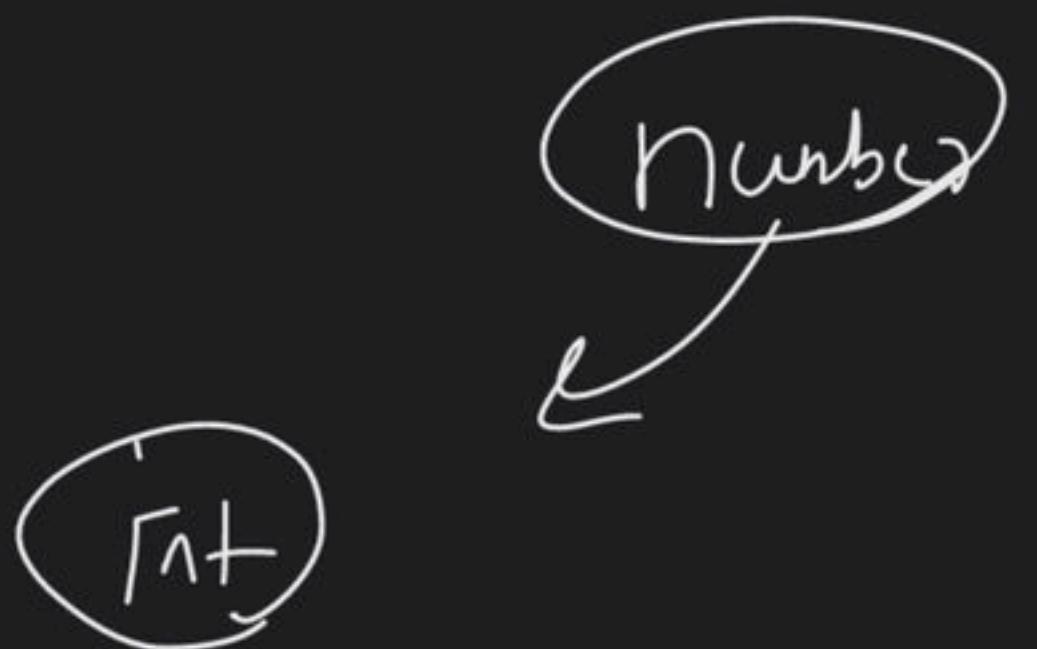
return 5

25 → 5

26 → 5 -

27 → 5 -

28 → 5 - -

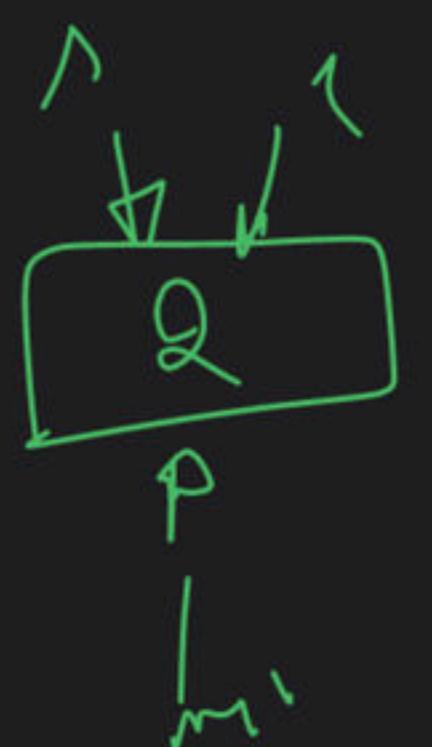


Int =

hot And like

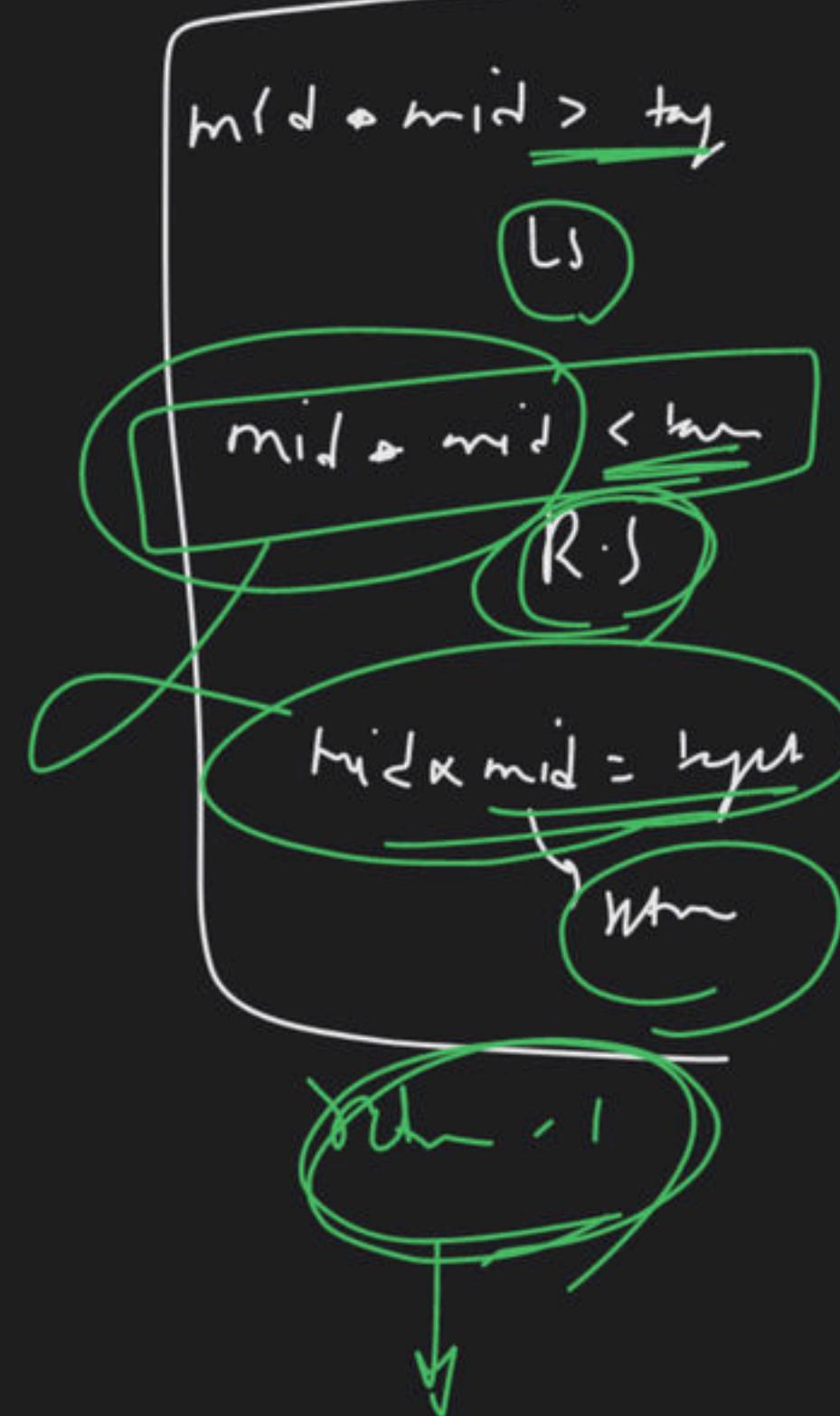
A diagram illustrating a relationship between a concept and a phrase. On the right, there is a circle containing the text "Reload". An arrow points from this circle to the phrase "hot And like" located above it.

$$|\Phi| = |<6$$



$$2 \oplus 2 > 1 < 8$$

$$\begin{array}{l} d = 3 \\ e = 2 \end{array} \rightarrow R_0 \text{ und } \underline{\underline{d'_e}}$$



Crust

Dry fun

2 - 3 cx

2 S

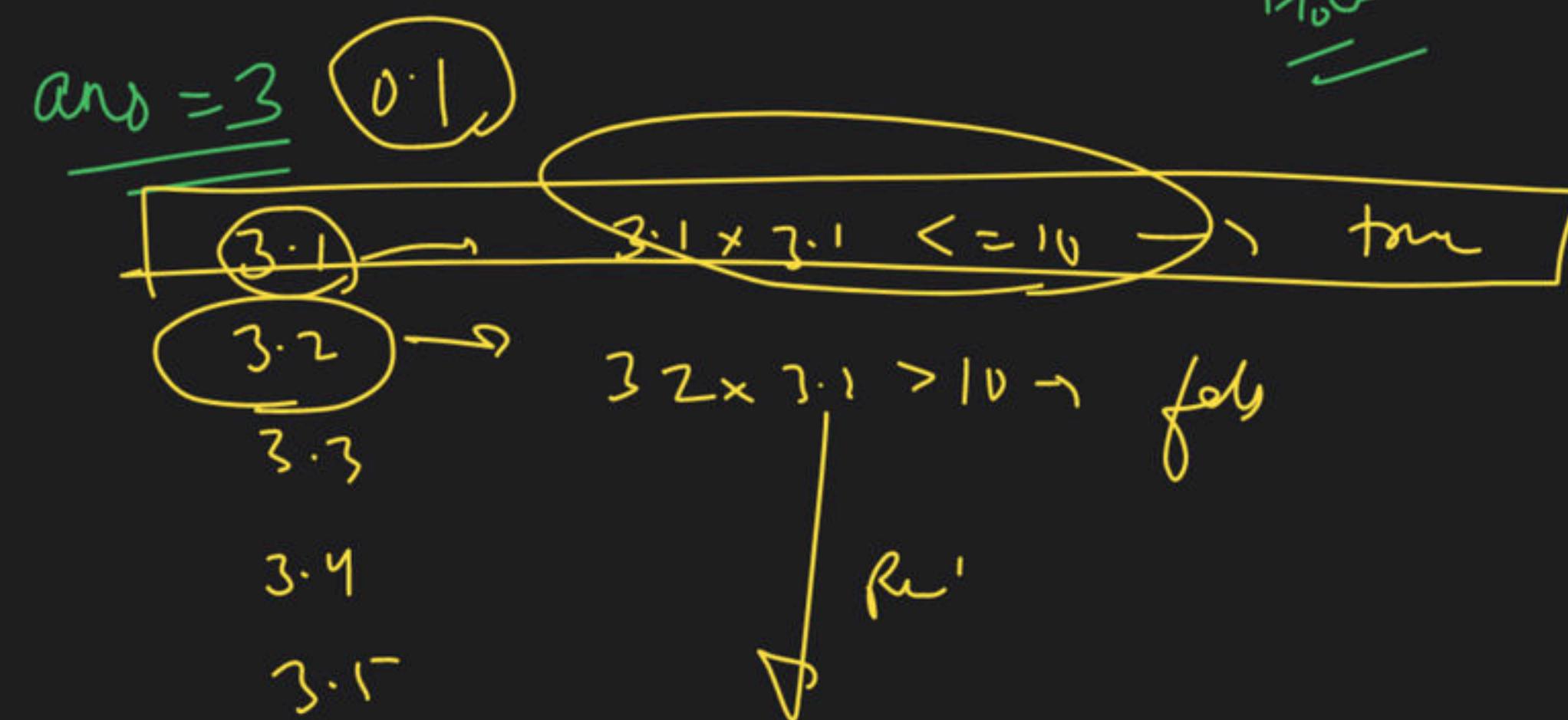
S

Spores

find float)

$$\sqrt{10} = 3.16$$

$\frac{3}{0.1}$   
 $\frac{0.1}{3.1}$   
 $\frac{0.1}{3.2}$   
 $\frac{0.1}{3.3}$



(0.01)

ans =  $\frac{3.1}{?}$

$3.11 \rightarrow 3.11 \times 3.11 <= 10$

$3.12 \rightarrow 3.11 \times 3.12 <= 10$

$3.13 \rightarrow 3.11 \times 3.13 <= 10$

$3.14 \rightarrow \dots$

$3.15 \rightarrow \dots$

$3.16 \rightarrow \dots$

$3.17 \rightarrow \text{false}$

3.18  
3.19

$$\begin{array}{r} 3 \cdot 1 \\ 0 \cdot 0 \cdot 1 \\ \hline 3 \cdot 1 \cdot 1 \\ 0 \cdot 0 \cdot 1 \\ \hline 3 \cdot 1 \cdot 2 \end{array}$$

3.16  
0.001  
~~3.161~~

3.161  
3.16L  
0.001

3.16S

3.119



Bunk for

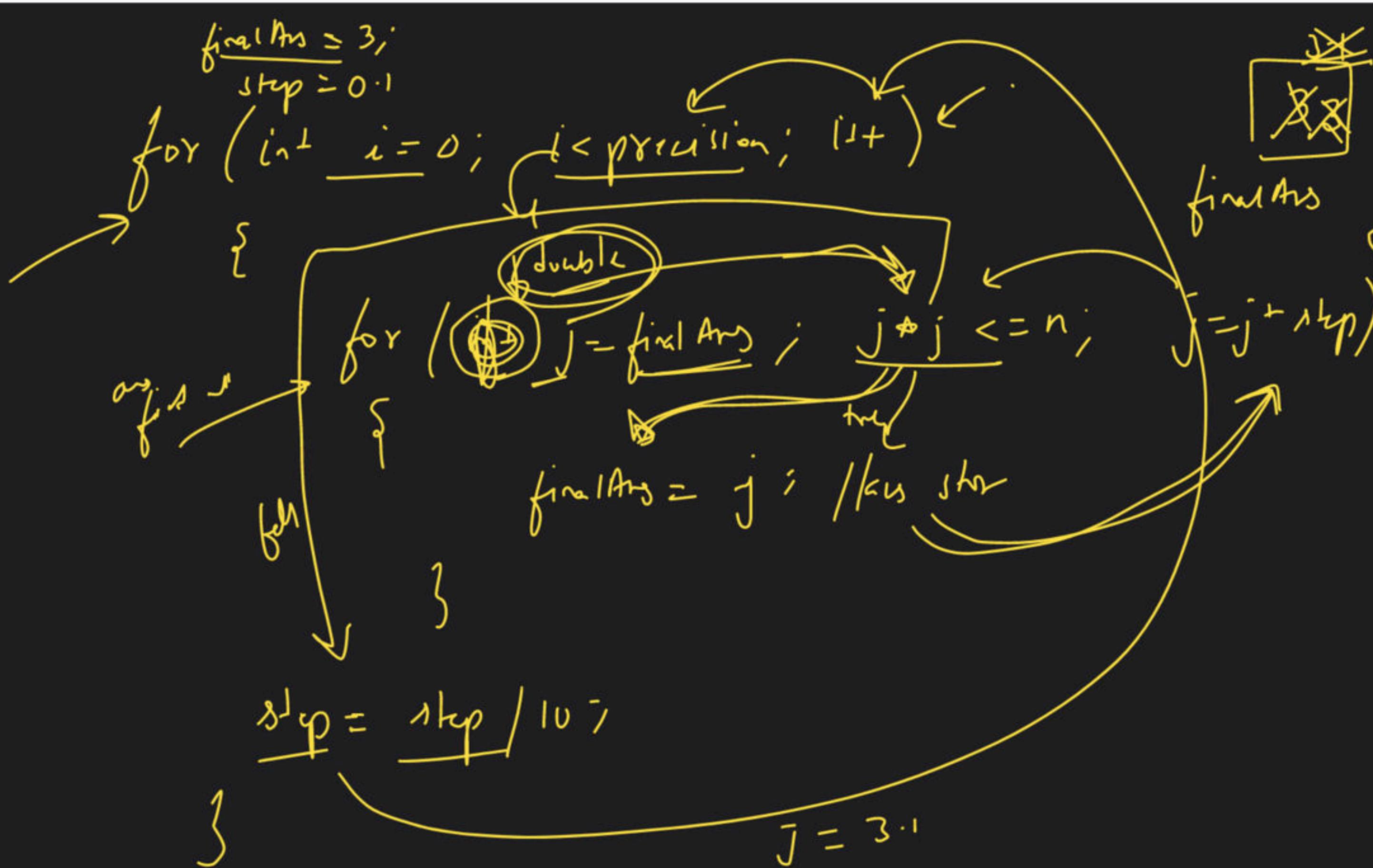
0.0001

0.00001

0.000001

-----

Binary search



$$j = 3.12 + 0.10 \\ = 3.13$$

$$\bar{j} = 3.11 + 0.01 \\ = 3.12$$

$$\bar{j} = 3.1 \times 3.1 = 9.61 \leq 10 \rightarrow \text{true}$$

$$\bar{j} = 3.1 + 0.01 \rightarrow 3.11$$

$$3.13 \times 3.13 \rightarrow 9.869 \\ \leftarrow 10 \quad 3.11 \times 3.11 = 9.621 \leq 10 \rightarrow \text{true}$$

~~3.11~~  
~~3.12~~  
~~3.13~~  
~~3.14~~  
~~3.15~~  
~~3.16~~  
~~3.17~~  
~~3.18~~  
~~3.19~~  
~~3.20~~  
~~3.21~~  
~~3.22~~  
~~3.23~~  
~~3.24~~  
~~3.25~~  
~~3.26~~  
~~3.27~~  
~~3.28~~  
~~3.29~~  
~~3.30~~  
~~3.31~~  
~~3.32~~  
~~3.33~~  
~~3.34~~  
~~3.35~~  
~~3.36~~  
~~3.37~~  
~~3.38~~  
~~3.39~~  
~~3.40~~  
~~3.41~~  
~~3.42~~  
~~3.43~~  
~~3.44~~  
~~3.45~~  
~~3.46~~  
~~3.47~~  
~~3.48~~  
~~3.49~~  
~~3.50~~  
~~3.51~~  
~~3.52~~  
~~3.53~~  
~~3.54~~  
~~3.55~~  
~~3.56~~  
~~3.57~~  
~~3.58~~  
~~3.59~~  
~~3.60~~  
~~3.61~~  
~~3.62~~  
~~3.63~~  
~~3.64~~  
~~3.65~~  
~~3.66~~  
~~3.67~~  
~~3.68~~  
~~3.69~~  
~~3.70~~  
~~3.71~~  
~~3.72~~  
~~3.73~~  
~~3.74~~  
~~3.75~~  
~~3.76~~  
~~3.77~~  
~~3.78~~  
~~3.79~~  
~~3.80~~  
~~3.81~~  
~~3.82~~  
~~3.83~~  
~~3.84~~  
~~3.85~~  
~~3.86~~  
~~3.87~~  
~~3.88~~  
~~3.89~~  
~~3.90~~  
~~3.91~~  
~~3.92~~  
~~3.93~~  
~~3.94~~  
~~3.95~~  
~~3.96~~  
~~3.97~~  
~~3.98~~  
~~3.99~~  
~~3.100~~  
~~3.101~~  
~~3.102~~  
~~3.103~~  
~~3.104~~  
~~3.105~~  
~~3.106~~  
~~3.107~~  
~~3.108~~  
~~3.109~~  
~~3.110~~  
~~3.111~~  
~~3.112~~  
~~3.113~~  
~~3.114~~  
~~3.115~~  
~~3.116~~  
~~3.117~~  
~~3.118~~  
~~3.119~~  
~~3.120~~  
~~3.121~~  
~~3.122~~  
~~3.123~~  
~~3.124~~  
~~3.125~~  
~~3.126~~  
~~3.127~~  
~~3.128~~  
~~3.129~~  
~~3.130~~  
~~3.131~~  
~~3.132~~  
~~3.133~~  
~~3.134~~  
~~3.135~~  
~~3.136~~  
~~3.137~~  
~~3.138~~  
~~3.139~~  
~~3.140~~  
~~3.141~~  
~~3.142~~  
~~3.143~~  
~~3.144~~  
~~3.145~~  
~~3.146~~  
~~3.147~~  
~~3.148~~  
~~3.149~~  
~~3.150~~  
~~3.151~~  
~~3.152~~  
~~3.153~~  
~~3.154~~  
~~3.155~~  
~~3.156~~  
~~3.157~~  
~~3.158~~  
~~3.159~~  
~~3.160~~  
~~3.161~~  
~~3.162~~  
~~3.163~~  
~~3.164~~  
~~3.165~~  
~~3.166~~  
~~3.167~~  
~~3.168~~  
~~3.169~~  
~~3.170~~  
~~3.171~~  
~~3.172~~  
~~3.173~~  
~~3.174~~  
~~3.175~~  
~~3.176~~  
~~3.177~~  
~~3.178~~  
~~3.179~~  
~~3.180~~  
~~3.181~~  
~~3.182~~  
~~3.183~~  
~~3.184~~  
~~3.185~~  
~~3.186~~  
~~3.187~~  
~~3.188~~  
~~3.189~~  
~~3.190~~  
~~3.191~~  
~~3.192~~  
~~3.193~~  
~~3.194~~  
~~3.195~~  
~~3.196~~  
~~3.197~~  
~~3.198~~  
~~3.199~~  
~~3.200~~  
~~3.201~~  
~~3.202~~  
~~3.203~~  
~~3.204~~  
~~3.205~~  
~~3.206~~  
~~3.207~~  
~~3.208~~  
~~3.209~~  
~~3.210~~  
~~3.211~~  
~~3.212~~  
~~3.213~~  
~~3.214~~  
~~3.215~~  
~~3.216~~  
~~3.217~~  
~~3.218~~  
~~3.219~~  
~~3.220~~  
~~3.221~~  
~~3.222~~  
~~3.223~~  
~~3.224~~  
~~3.225~~  
~~3.226~~  
~~3.227~~  
~~3.228~~  
~~3.229~~  
~~3.230~~  
~~3.231~~  
~~3.232~~  
~~3.233~~  
~~3.234~~  
~~3.235~~  
~~3.236~~  
~~3.237~~  
~~3.238~~  
~~3.239~~  
~~3.240~~  
~~3.241~~  
~~3.242~~  
~~3.243~~  
~~3.244~~  
~~3.245~~  
~~3.246~~  
~~3.247~~  
~~3.248~~  
~~3.249~~  
~~3.250~~  
~~3.251~~  
~~3.252~~  
~~3.253~~  
~~3.254~~  
~~3.255~~  
~~3.256~~  
~~3.257~~  
~~3.258~~  
~~3.259~~  
~~3.260~~  
~~3.261~~  
~~3.262~~  
~~3.263~~  
~~3.264~~  
~~3.265~~  
~~3.266~~  
~~3.267~~  
~~3.268~~  
~~3.269~~  
~~3.270~~  
~~3.271~~  
~~3.272~~  
~~3.273~~  
~~3.274~~  
~~3.275~~  
~~3.276~~  
~~3.277~~  
~~3.278~~  
~~3.279~~  
~~3.280~~  
~~3.281~~  
~~3.282~~  
~~3.283~~  
~~3.284~~  
~~3.285~~  
~~3.286~~  
~~3.287~~  
~~3.288~~  
~~3.289~~  
~~3.290~~  
~~3.291~~  
~~3.292~~  
~~3.293~~  
~~3.294~~  
~~3.295~~  
~~3.296~~  
~~3.297~~  
~~3.298~~  
~~3.299~~  
~~3.300~~  
~~3.301~~  
~~3.302~~  
~~3.303~~  
~~3.304~~  
~~3.305~~  
~~3.306~~  
~~3.307~~  
~~3.308~~  
~~3.309~~  
~~3.310~~  
~~3.311~~  
~~3.312~~  
~~3.313~~  
~~3.314~~  
~~3.315~~  
~~3.316~~  
~~3.317~~  
~~3.318~~  
~~3.319~~  
~~3.320~~  
~~3.321~~  
~~3.322~~  
~~3.323~~  
~~3.324~~  
~~3.325~~  
~~3.326~~  
~~3.327~~  
~~3.328~~  
~~3.329~~  
~~3.330~~  
~~3.331~~  
~~3.332~~  
~~3.333~~  
~~3.334~~  
~~3.335~~  
~~3.336~~  
~~3.337~~  
~~3.338~~  
~~3.339~~  
~~3.340~~  
~~3.341~~  
~~3.342~~  
~~3.343~~  
~~3.344~~  
~~3.345~~  
~~3.346~~  
~~3.347~~  
~~3.348~~  
~~3.349~~  
~~3.350~~  
~~3.351~~  
~~3.352~~  
~~3.353~~  
~~3.354~~  
~~3.355~~  
~~3.356~~  
~~3.357~~  
~~3.358~~  
~~3.359~~  
~~3.360~~  
~~3.361~~  
~~3.362~~  
~~3.363~~  
~~3.364~~  
~~3.365~~  
~~3.366~~  
~~3.367~~  
~~3.368~~  
~~3.369~~  
~~3.370~~  
~~3.371~~  
~~3.372~~  
~~3.373~~  
~~3.374~~  
~~3.375~~  
~~3.376~~  
~~3.377~~  
~~3.378~~  
~~3.379~~  
~~3.380~~  
~~3.381~~  
~~3.382~~  
~~3.383~~  
~~3.384~~  
~~3.385~~  
~~3.386~~  
~~3.387~~  
~~3.388~~  
~~3.389~~  
~~3.390~~  
~~3.391~~  
~~3.392~~  
~~3.393~~  
~~3.394~~  
~~3.395~~  
~~3.396~~  
~~3.397~~  
~~3.398~~  
~~3.399~~  
~~3.400~~  
~~3.401~~  
~~3.402~~  
~~3.403~~  
~~3.404~~  
~~3.405~~  
~~3.406~~  
~~3.407~~  
~~3.408~~  
~~3.409~~  
~~3.410~~  
~~3.411~~  
~~3.412~~  
~~3.413~~  
~~3.414~~  
~~3.415~~  
~~3.416~~  
~~3.417~~  
~~3.418~~  
~~3.419~~  
~~3.420~~  
~~3.421~~  
~~3.422~~  
~~3.423~~  
~~3.424~~  
~~3.425~~  
~~3.426~~  
~~3.427~~  
~~3.428~~  
~~3.429~~  
~~3.430~~  
~~3.431~~  
~~3.432~~  
~~3.433~~  
~~3.434~~  
~~3.435~~  
~~3.436~~  
~~3.437~~  
~~3.438~~  
~~3.439~~  
~~3.440~~  
~~3.441~~  
~~3.442~~  
~~3.443~~  
~~3.444~~  
~~3.445~~  
~~3.446~~  
~~3.447~~  
~~3.448~~  
~~3.449~~  
~~3.450~~  
~~3.451~~  
~~3.452~~  
~~3.453~~  
~~3.454~~  
~~3.455~~  
~~3.456~~  
~~3.457~~  
~~3.458~~  
~~3.459~~  
~~3.460~~  
~~3.461~~  
~~3.462~~  
~~3.463~~  
~~3.464~~  
~~3.465~~  
~~3.466~~  
~~3.467~~  
~~3.468~~  
~~3.469~~  
~~3.470~~  
~~3.471~~  
~~3.472~~  
~~3.473~~  
~~3.474~~  
~~3.475~~  
~~3.476~~  
~~3.477~~  
~~3.478~~  
~~3.479~~  
~~3.480~~  
~~3.481~~  
~~3.482~~  
~~3.483~~  
~~3.484~~  
~~3.485~~  
~~3.486~~  
~~3.487~~  
~~3.488~~  
~~3.489~~  
~~3.490~~  
~~3.491~~  
~~3.492~~  
~~3.493~~  
~~3.494~~  
~~3.495~~  
~~3.496~~  
~~3.497~~  
~~3.498~~  
~~3.499~~  
~~3.500~~  
~~3.501~~  
~~3.502~~  
~~3.503~~  
~~3.504~~  
~~3.505~~  
~~3.506~~  
~~3.507~~  
~~3.508~~  
~~3.509~~  
~~3.510~~  
~~3.511~~  
~~3.512~~  
~~3.513~~  
~~3.514~~  
~~3.515~~  
~~3.516~~  
~~3.517~~  
~~3.518~~  
~~3.519~~  
~~3.520~~  
~~3.521~~  
~~3.522~~  
~~3.523~~  
~~3.524~~  
~~3.525~~  
~~3.526~~  
~~3.527~~  
~~3.528~~  
~~3.529~~  
~~3.530~~  
~~3.531~~  
~~3.532~~  
~~3.533~~  
~~3.534~~  
~~3.535~~  
~~3.536~~  
~~3.537~~  
~~3.538~~  
~~3.539~~  
~~3.540~~  
~~3.541~~  
~~3.542~~  
~~3.543~~  
~~3.544~~  
~~3.545~~  
~~3.546~~  
~~3.547~~  
~~3.548~~  
~~3.549~~  
~~3.550~~  
~~3.551~~  
~~3.552~~  
~~3.553~~  
~~3.554~~  
~~3.555~~  
~~3.556~~  
~~3.557~~  
~~3.558~~  
~~3.559~~  
~~3.560~~  
~~3.561~~  
~~3.562~~  
~~3.563~~  
~~3.564~~  
~~3.565~~  
~~3.566~~  
~~3.567~~  
~~3.568~~  
~~3.569~~  
~~3.570~~  
~~3.571~~  
<

```
for ( int i = 0 ; !isPrecision ; i++ )
```

    Research  
    30.

```
    for ( double j = findAns ; j < j <= n ; j += step )
```

```
    {
```

```
        finalAns = j;
```

```
    }
```

```
    step = step / 10;
```

```
}
```

7 7 7 7 + 2

7 7 7 7 12

Binary Search

$i, j \rightarrow c * i + j$   
2D matrix

arr  
rowIndex  
colIndex  
 $\uparrow$

mid  
cols

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16
17	18	19	20

$c * i + j$   
if arr[rowIndex][colIndex] < target  
R.S  
L.S

target = 11  
 $n \rightarrow \text{rows}$   
 $m \rightarrow \text{cols}$

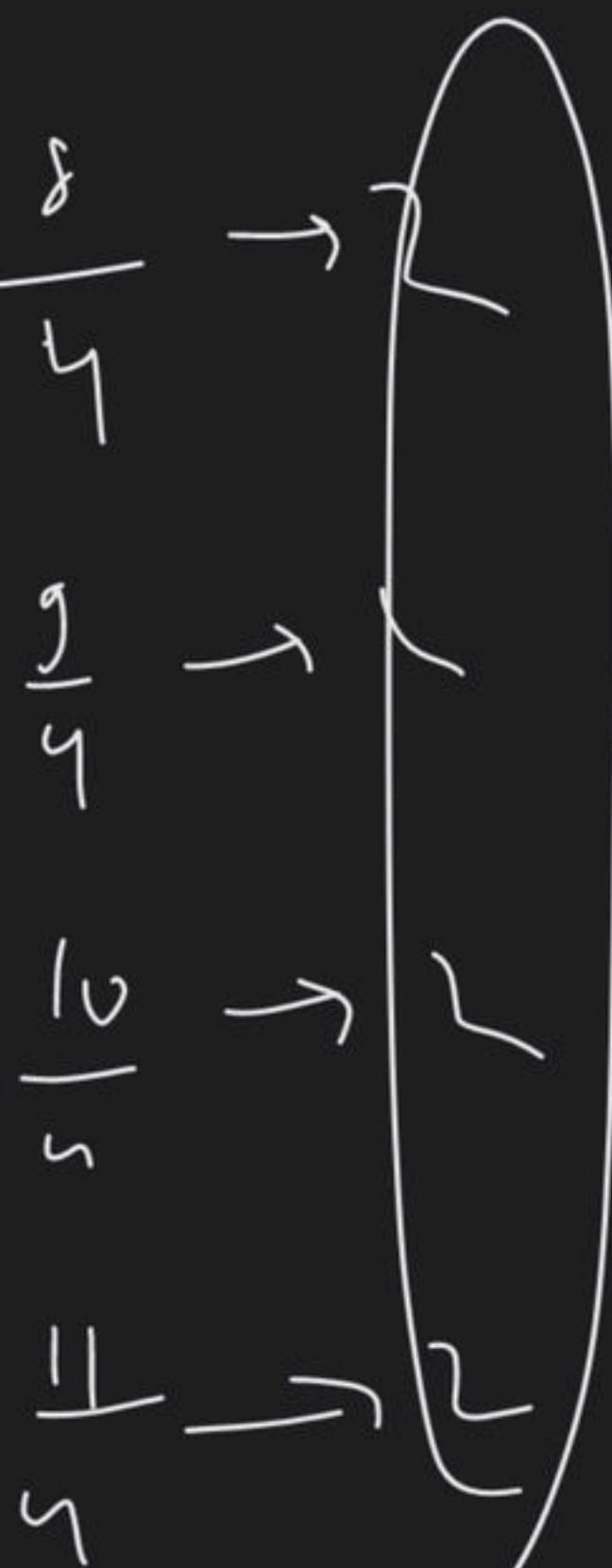
$i = 0$   
 $e = n * m - 1$   
 $= 20 - 1$   
 $\text{mid} = \frac{0 + 19}{2} = 9$

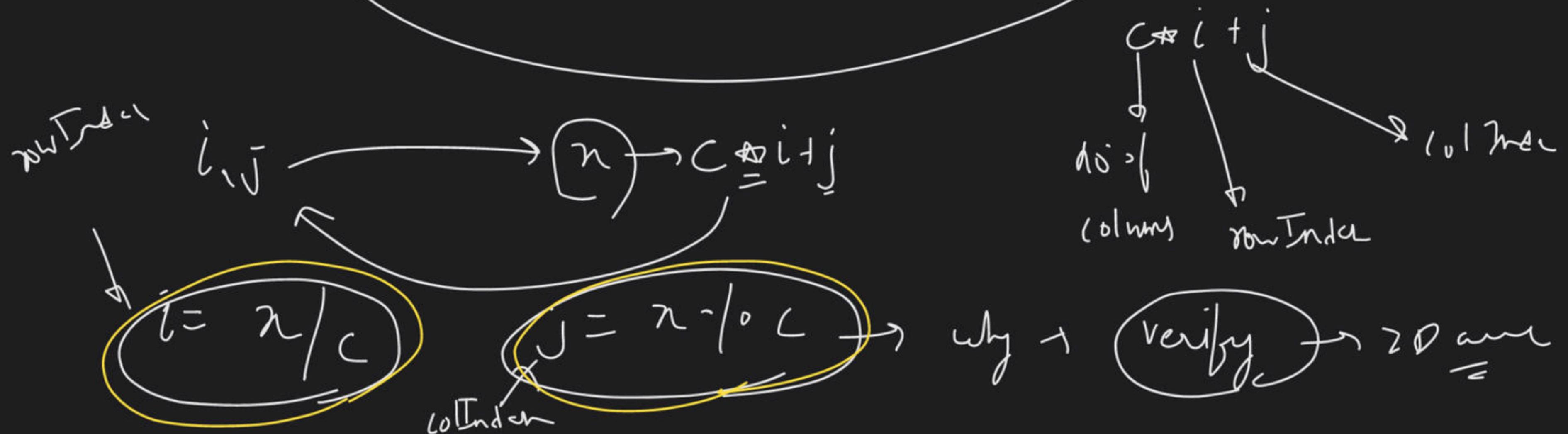
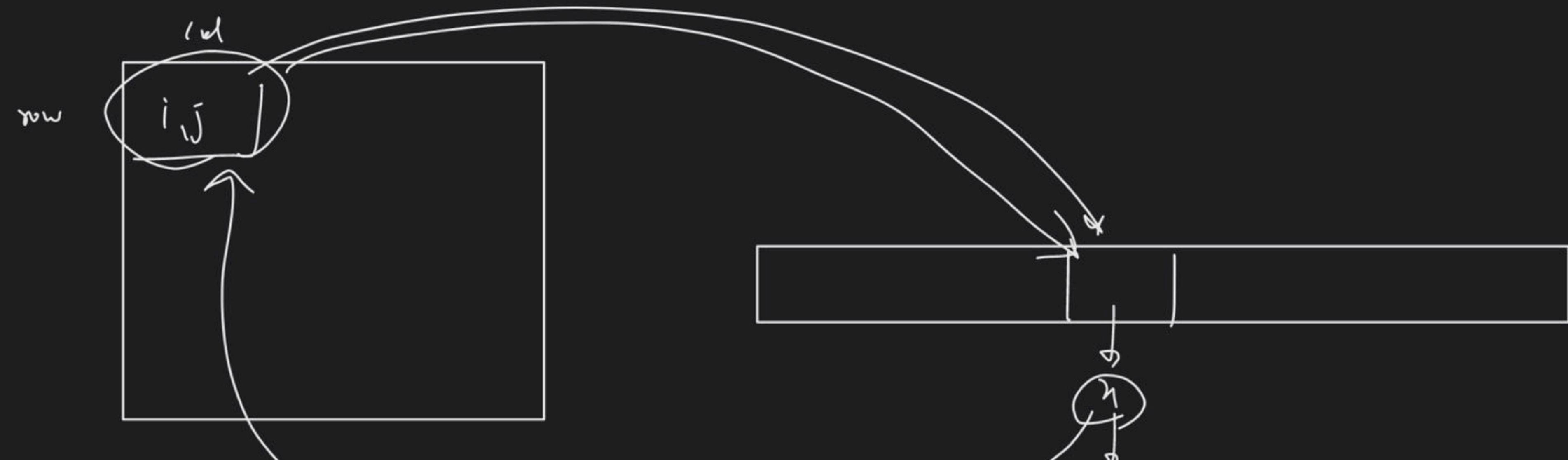
$\text{rowIndex} = \frac{\text{mid}}{\text{cols}} = \frac{9}{4} = 2$   
 $\text{colIndex} = \text{mid} \% \text{cols} = 9 \% 4 = 1$

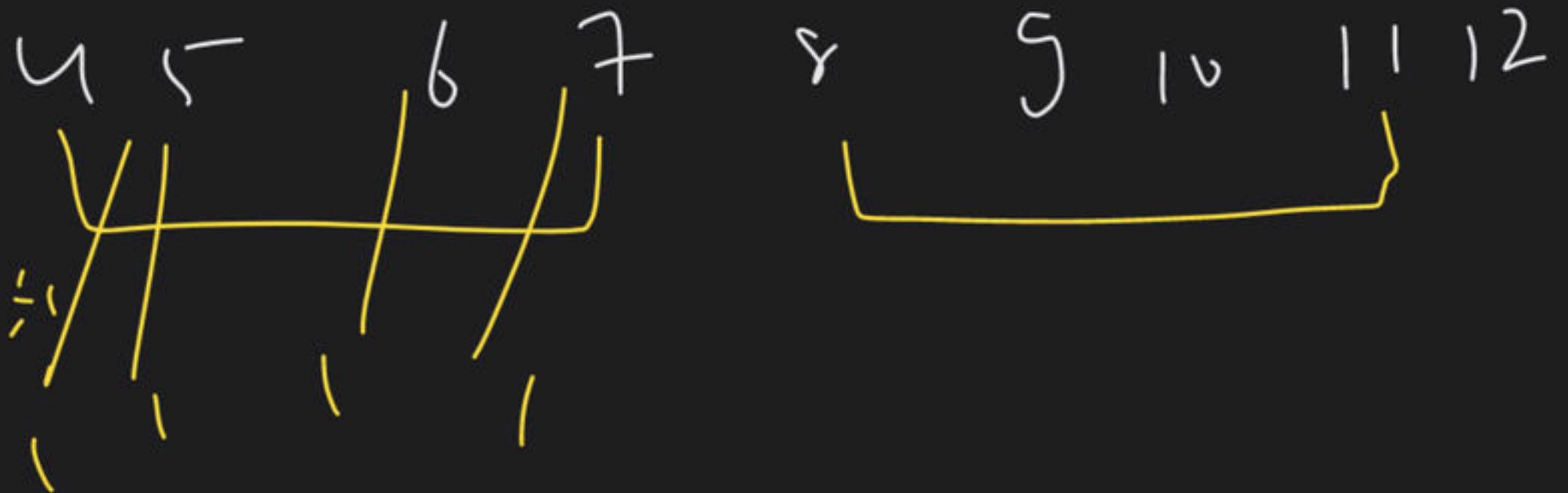
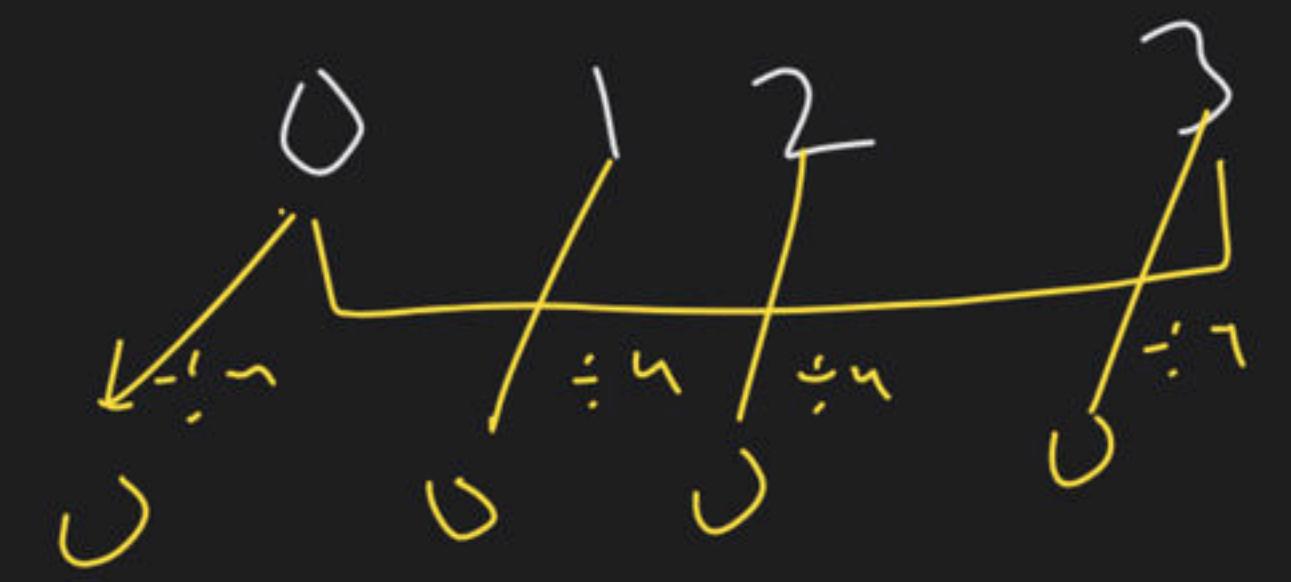
```

int s = 0;           mid
int totalSize =  $\lceil \frac{m}{2} \rceil$  - 1
int e = totalSize - 1
int mid =  $\frac{s+e}{2}$ ;
while (s <= e)
{
    rowIndex = mid / cols =  $\lceil \frac{mid}{m} \rceil$ ;
    colIndex = mid % m;
    int element = arr[rowIndex][colIndex];
    if (element == target)
        return true;
    if (element > target)
        left)  $\rightarrow e = mid - 1$ 
    else
        RS  $\rightarrow s = mid + 1$ 
}

```









- ① Search in a nearly sorted array in  $\log$  time
- ② Division of 2 rd using Binary Search
- ③ find odd occurring element using Binary Search





