



# Heap Class-4

Special class

→ 1 min → focus

[4 3 6 7]

$K=3$

7 -  $\frac{7}{2}$

4

6 -  $\frac{6}{2}$

4 -  $\frac{4}{2}$

[4 3 6 4]

$K=2$

[4 3 3 4]

$K=1$

[2 3 3 4]

$K=0$

$2 + 3 + 3 + 4 = 12$

$$K = -2$$

min no

$$\text{max heap} \rightarrow (5) - \frac{5}{2}$$

fetch  $\rightarrow 5$

process  $\rightarrow 5 - \frac{5}{2} 2^3$

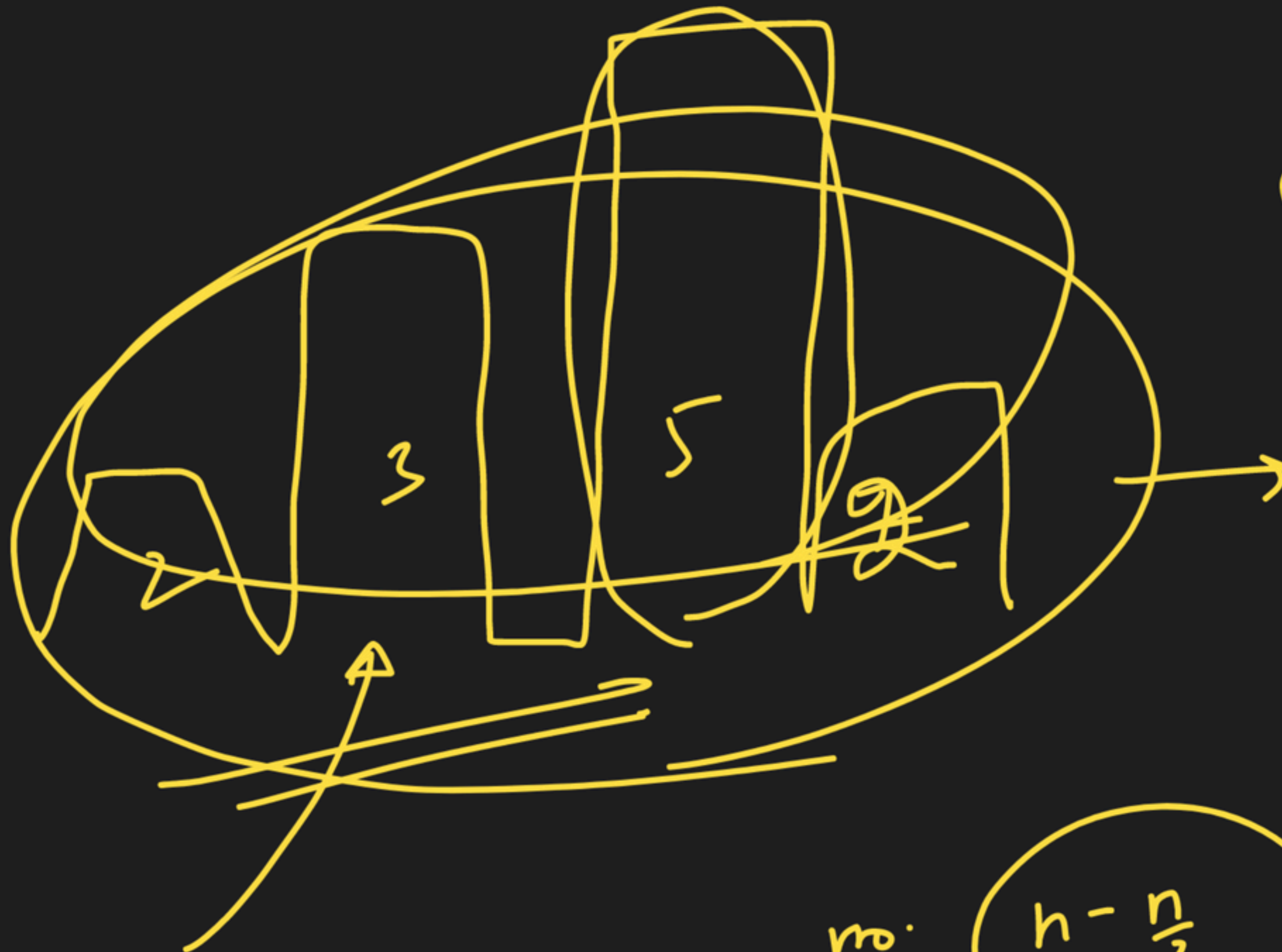
put = 2 put

$$K! = 0$$

no.

$$h = \frac{n}{2}$$

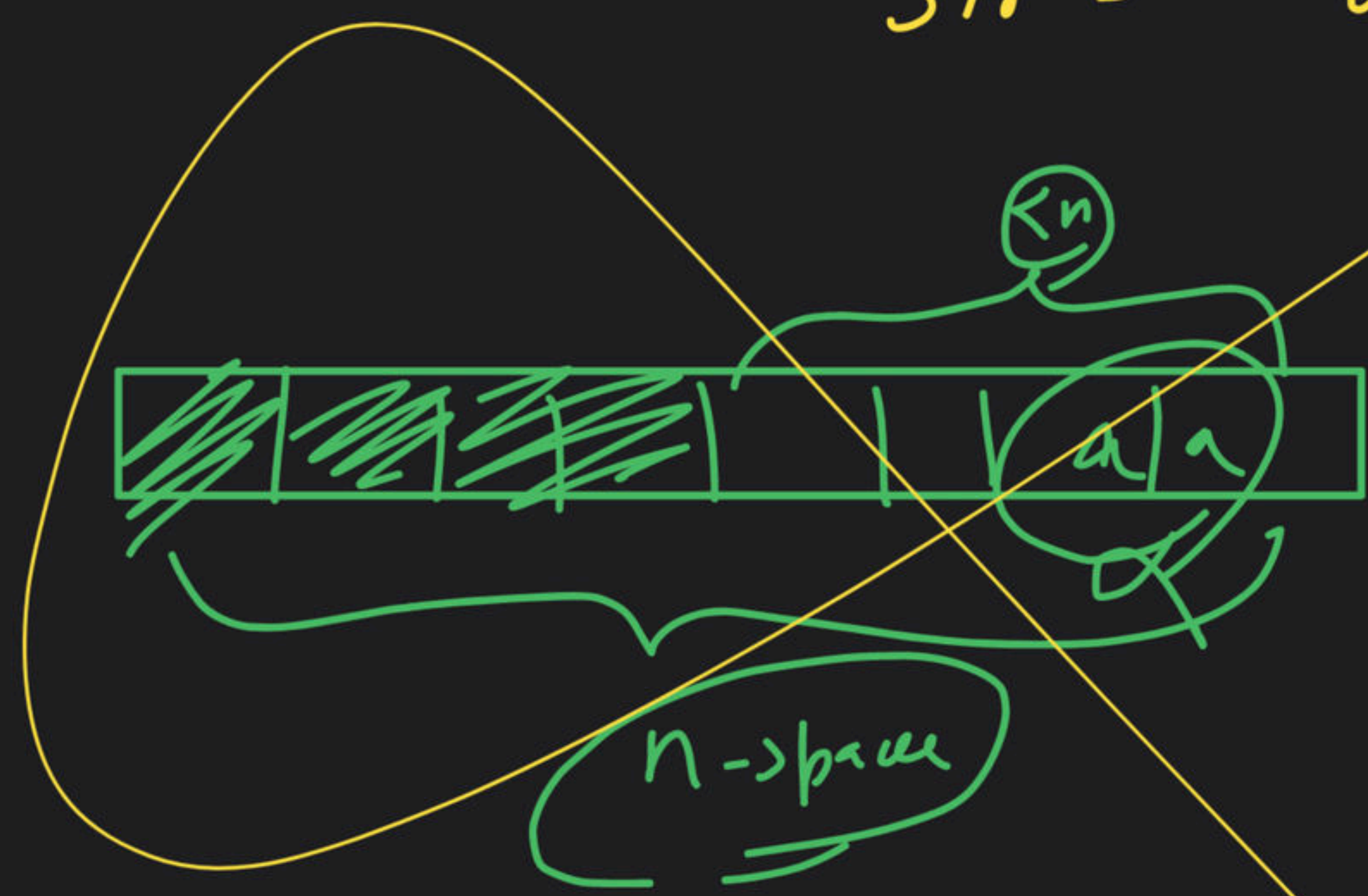
$$3 - \frac{3}{2}$$





# → Reorganise Strings

str = "aabbbaab cd"



<del>a</del>	→	<del>4</del>
b	→	3
c	→	1
<del>d</del>	→	1

Total  
9

a → 4  
b → 3  
c → 1  
d → 1









→ Longest Happy String:-

$a \rightarrow \underline{3}$

$b \rightarrow \underline{4}$

$c \rightarrow \underline{7}$

~~$c \rightarrow \times$~~   ~~$\times$~~   ~~$\times$~~   ~~$\times$~~

~~$b \rightarrow \times$~~   ~~$\times$~~   $0$

~~$a \rightarrow \times$~~   ~~$\times$~~   $0$

✓  
✓

$a \rightarrow 'a' \text{ times}$   
 $b \rightarrow 'b' \text{ times}$   
 $c \rightarrow 'c' \text{ times}$  (almost)

$s \rightarrow$ 

	b	
--	---	--

~~aa~~

b b b

c c c

ccbbccaaaccbbac  
^   ^   ^

# → Median in a Stream:-

$$(n \log n) = \infty$$

i/p →

2, 4, 5, 3, 1, 0, 1, 1, 6, 18, 12

odd

$$\frac{n+1}{2}$$

even

00

2, 3, 5

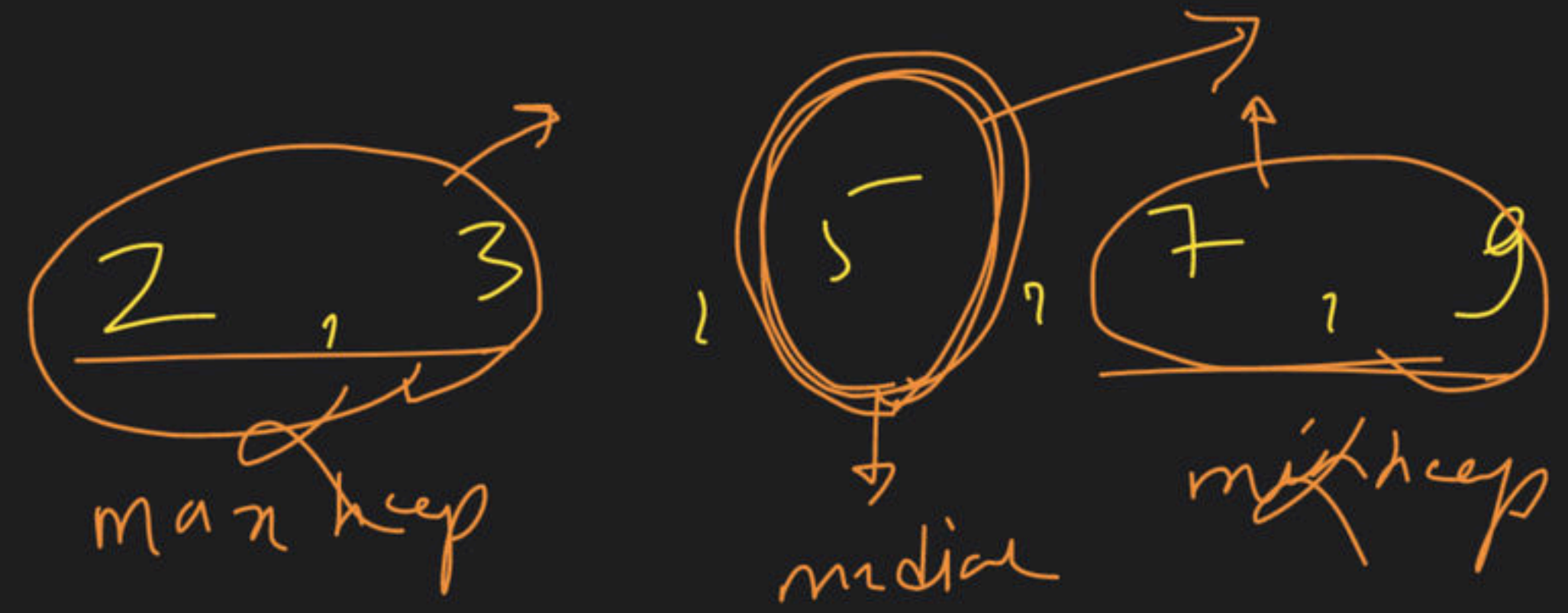
mid

2, 3, 5, 6

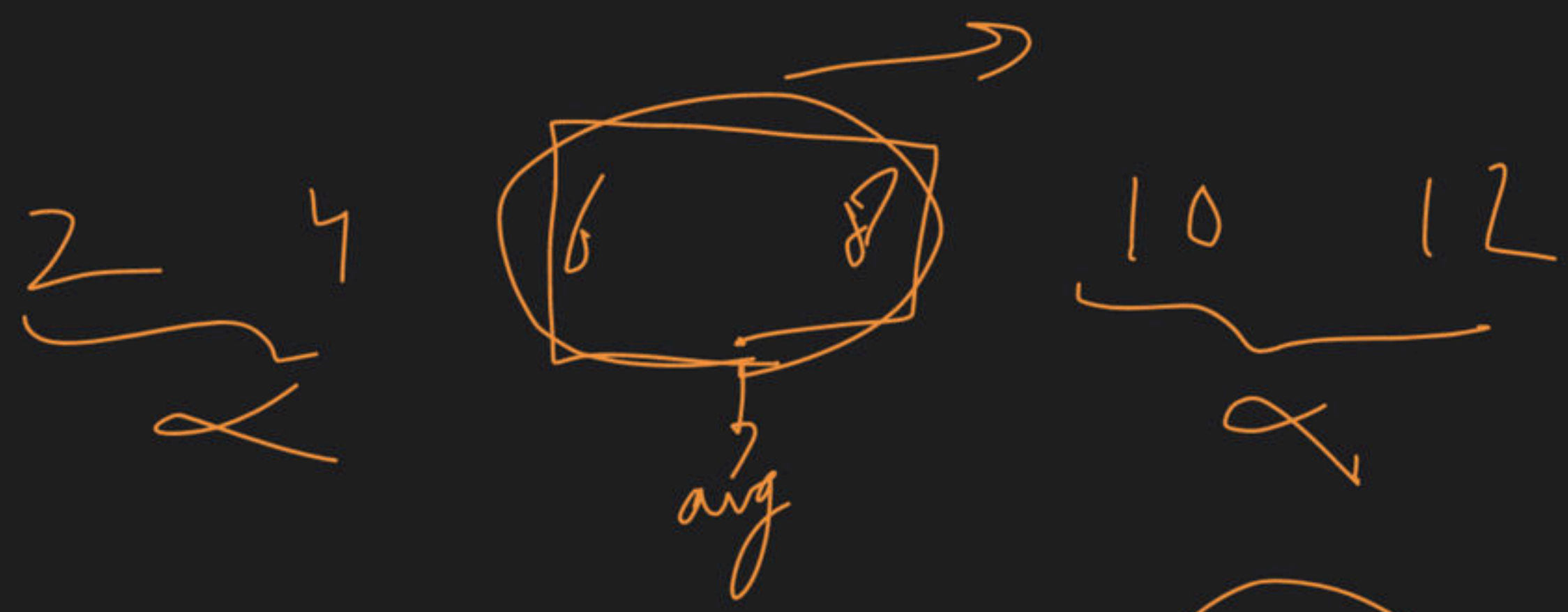
$$\frac{7+5}{2} = 4$$



Odd →



even →



$$\frac{6+8}{2} = \frac{14}{2} = 7$$



i/p  $\rightarrow$

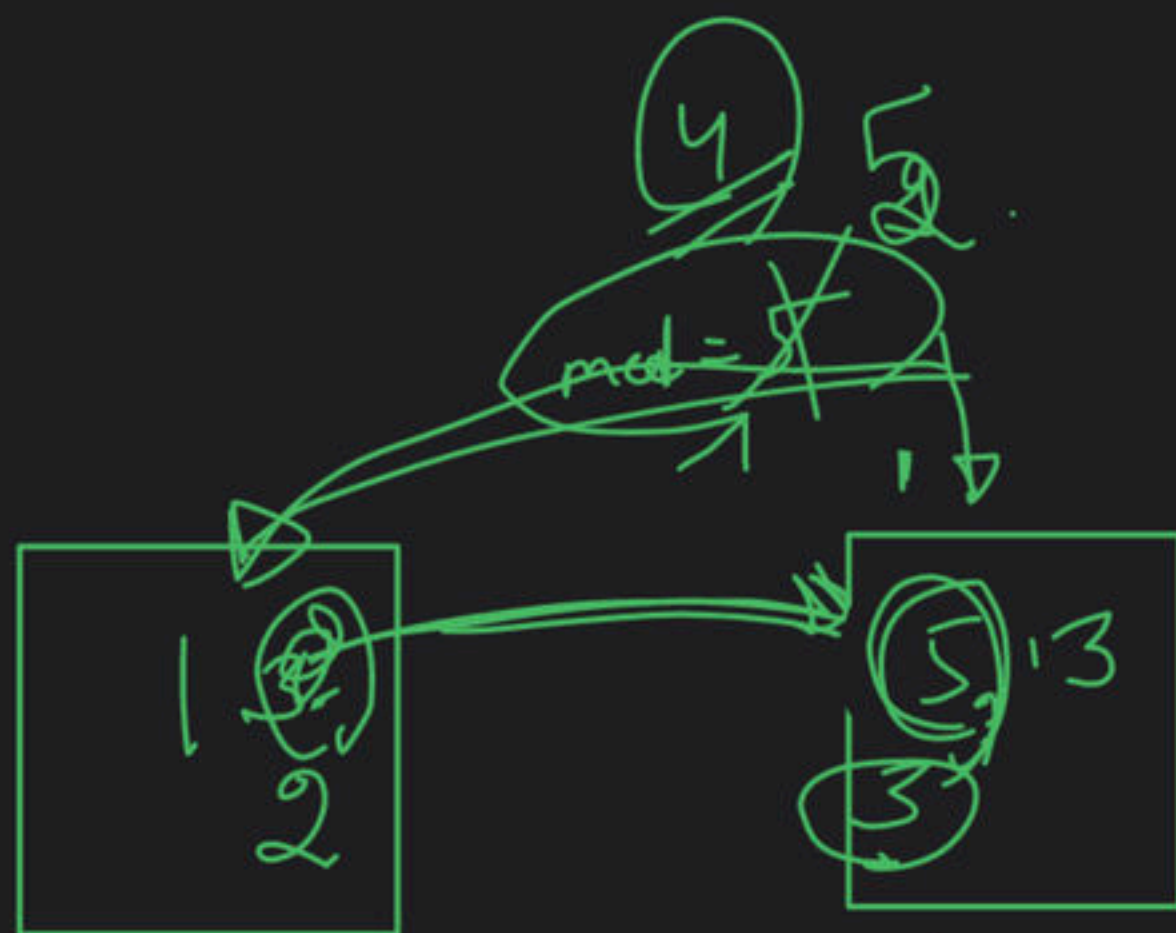
5, 15, 1, 3, 2, 8

o/p  $\rightarrow$  ?

Algo

2 min  
Break

5, 13, 1, 3, 2, 8



max

~~size = 3~~

min

~~size = 3~~

minheap = max + 1  
size

no < Brackets

↓ put max  
↓ avg

5 1 13 1 5 13

median = 0

max size == min size  
↓  
no > median → max put

max size = min size + 1  
↓  
no > median → minheap put  
↓  
median = avg

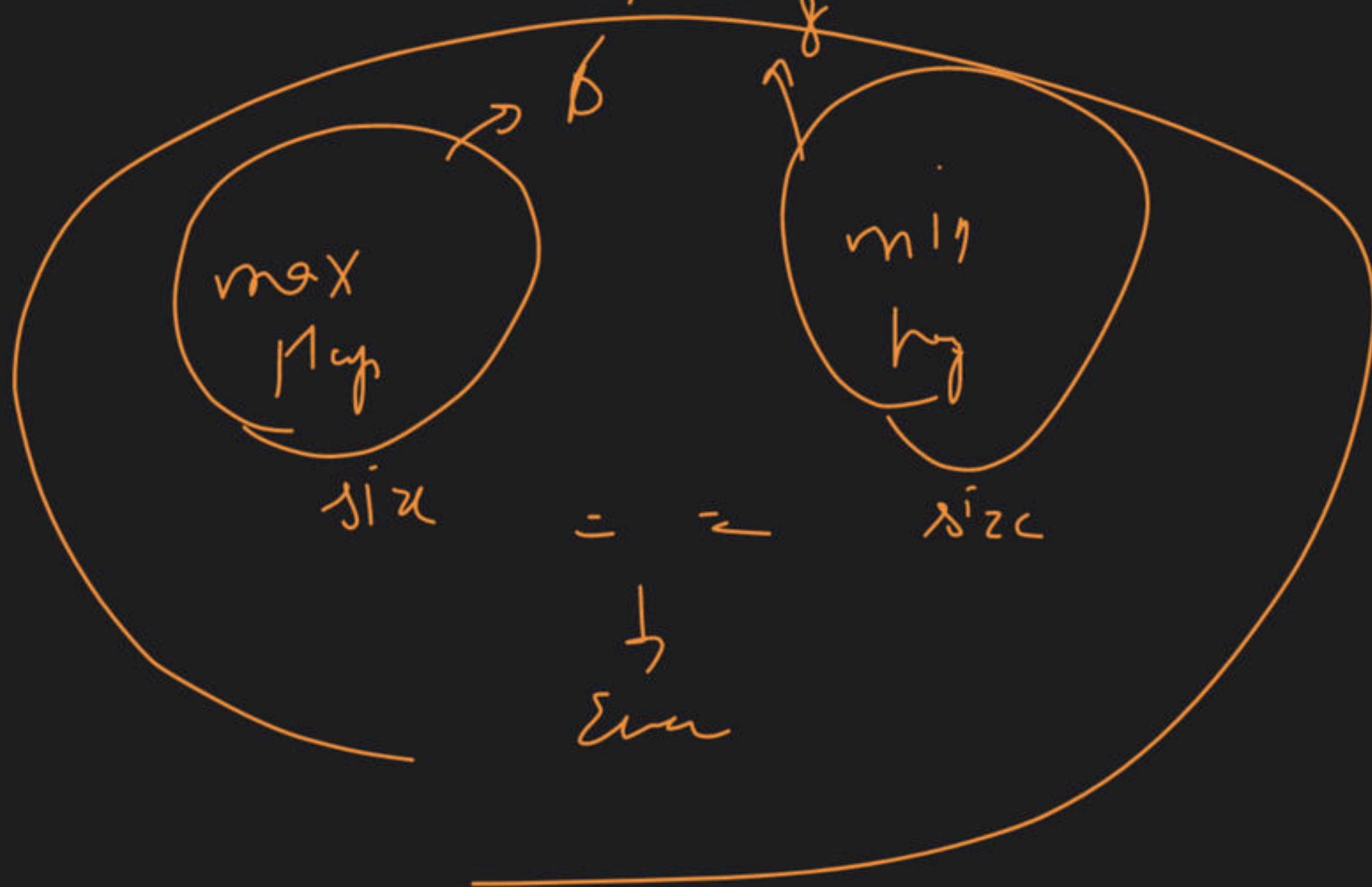
max size == min size

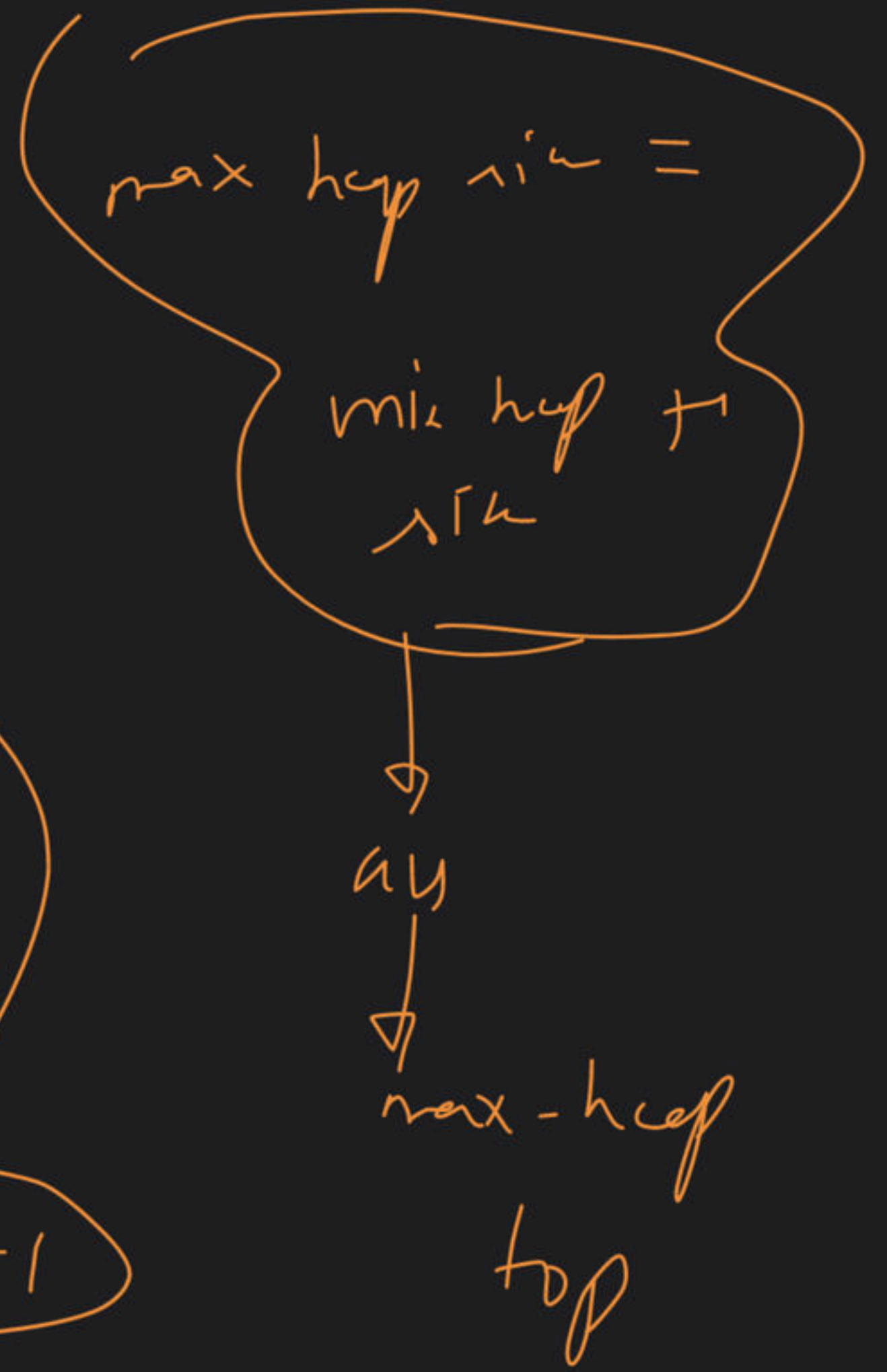
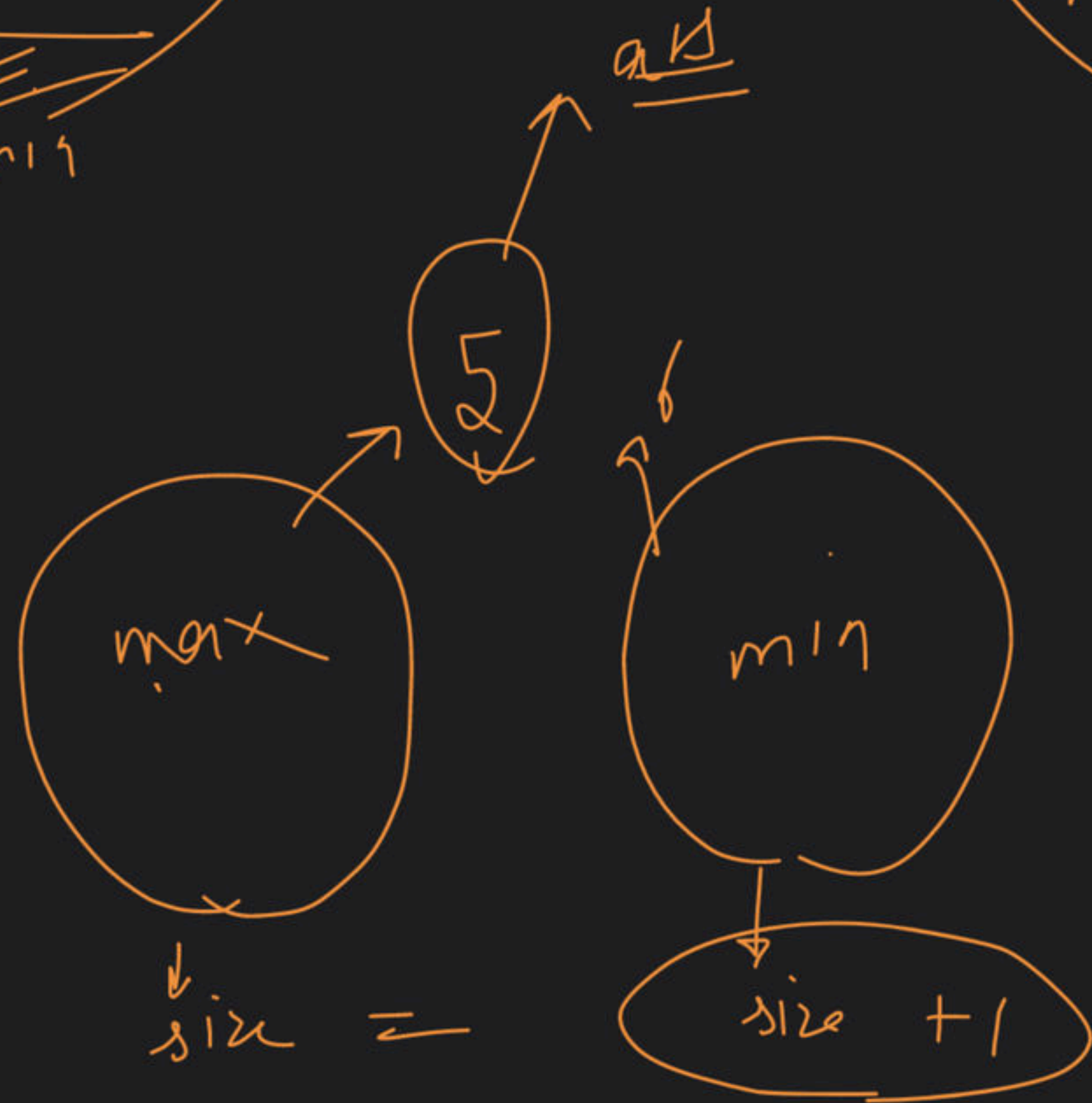
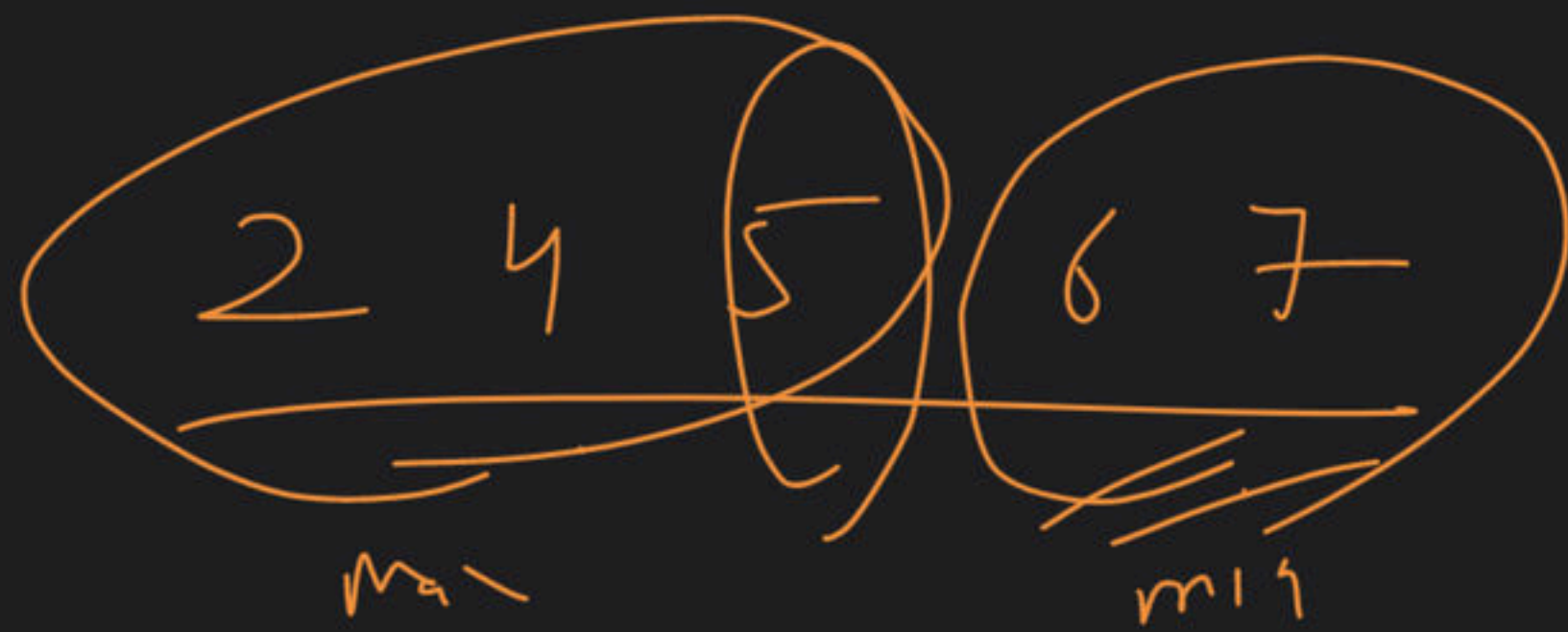
↓  
no < median → max Top → min  
no → max





avg  $\rightarrow \frac{6+8}{2} = \frac{14}{2} = 7$

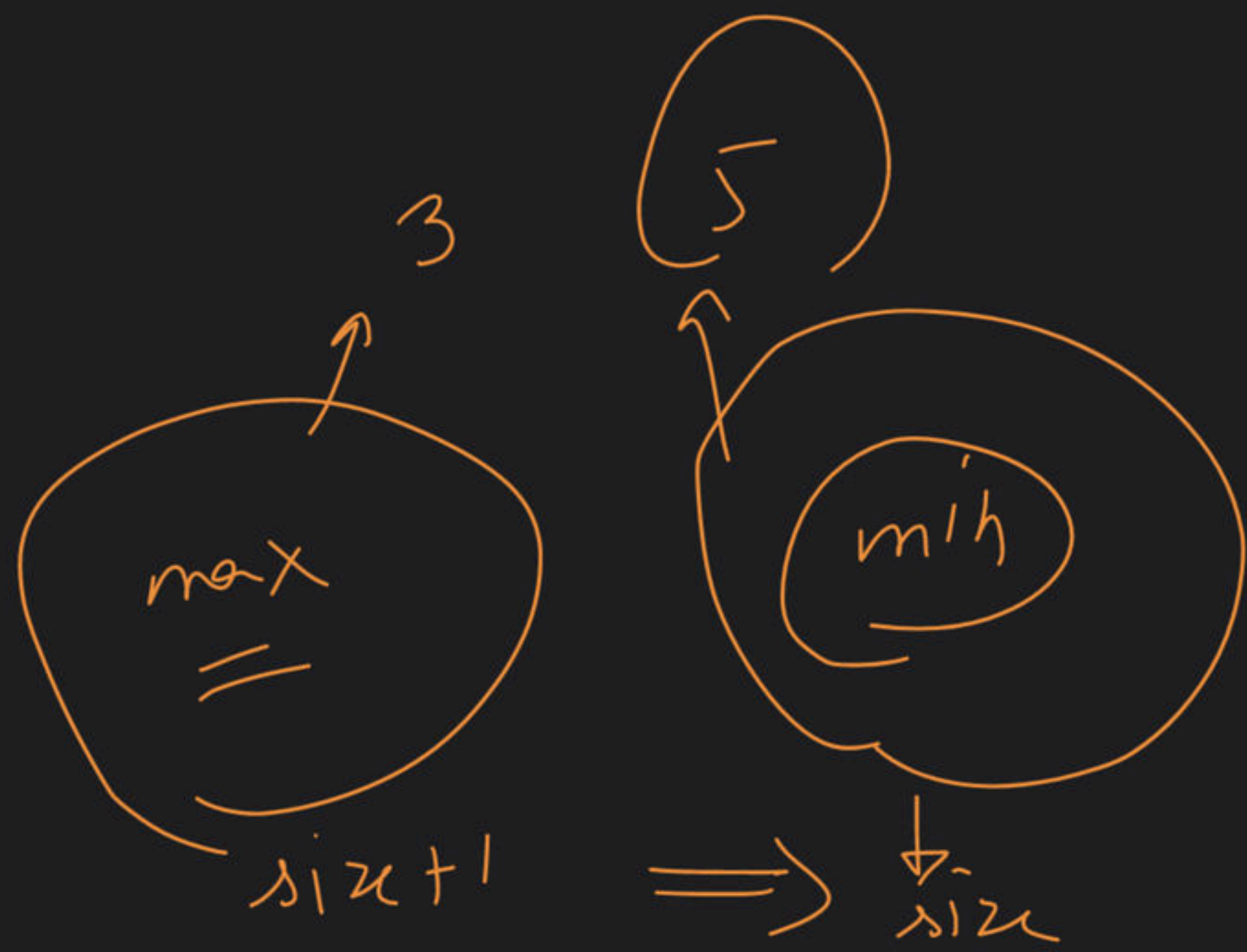








min heap size =  
max heap + 1  
size



↓  
ans  
↓  
min heap  
top

avg  $\rightarrow$  maxTop  
minTop

max  
heap  
size  
4

=

min  
heap  
size  
4

diff = 0

avg  $\rightarrow$  maxTop

max  
heap  
size  
4

=

min  
heap  
size  
3

+ 1

diff = 1

avg  $\rightarrow$  minTop

min  
heap  
size  
5

=

max  
heap  
size  
5

+ 1

diff = 1



Obs: -

→ max size = min size

↓  
✓ median →  $\text{avg}(\text{maxTop}, \text{minTop})$

→ max size = min size + 1

✓ median → (return maxTop)

→ min size = max size + 1

↓  
✓ median → (minTop)

insertion

↳  $\text{max size} == \text{min size}$  →  $\text{no} > \text{median}$  → push minHeap(no)

→  $\text{no} < \text{median}$  → push (maxHeap(no))

→  $\text{max size} == \text{min size} + 1$

→  $\text{no} > \text{median}$  → push (minHeap)

→  $\text{no} < \text{median}$  → I  $\text{maxTop} \rightarrow \text{min}$   
→ II  $\text{no} \rightarrow \text{max}$

→  $\text{min size} = \text{max size} + 1$

→  $\text{no} > \text{median}$  → max heap

→  $\text{no} < \text{median}$  → minTop → max  
→  $\text{no} \rightarrow \text{min}$



# insertion

min heap size  $\leq$  max heap size + 1

no > median

minTop  $\rightarrow$  max  
element  $\rightarrow$  min

no < median

maxi-pull

no = 25

no = 2

med = 5

