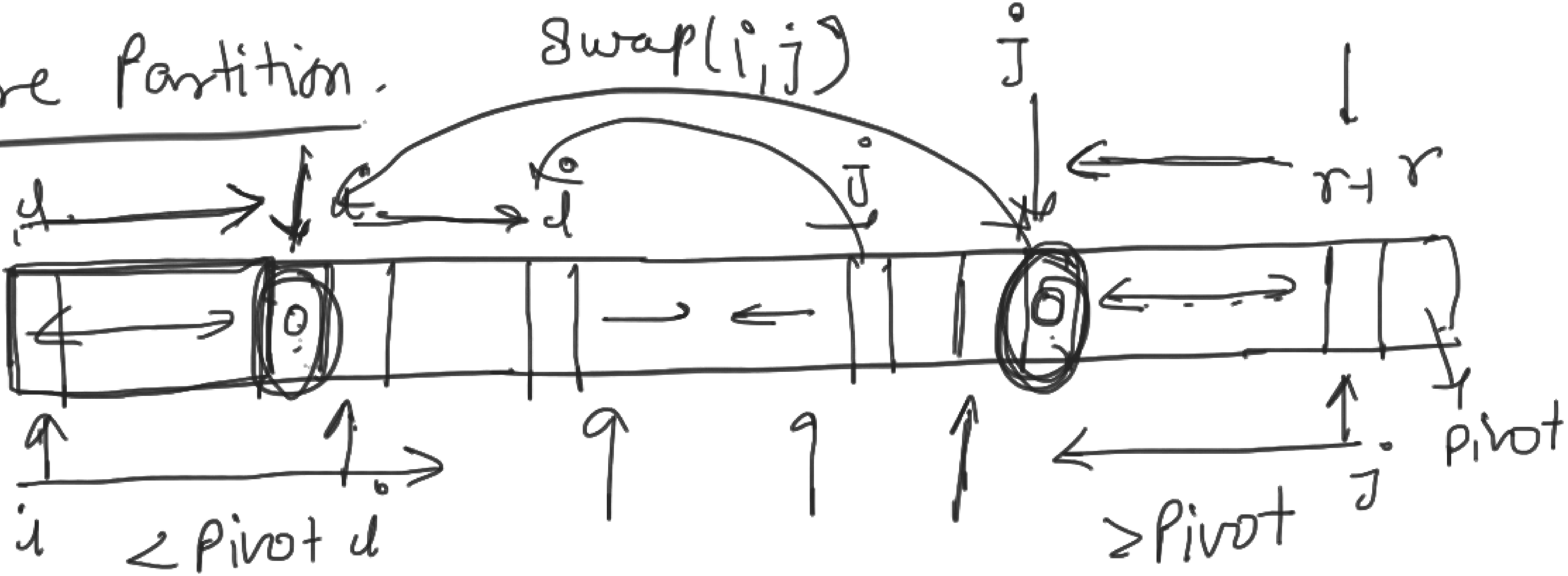
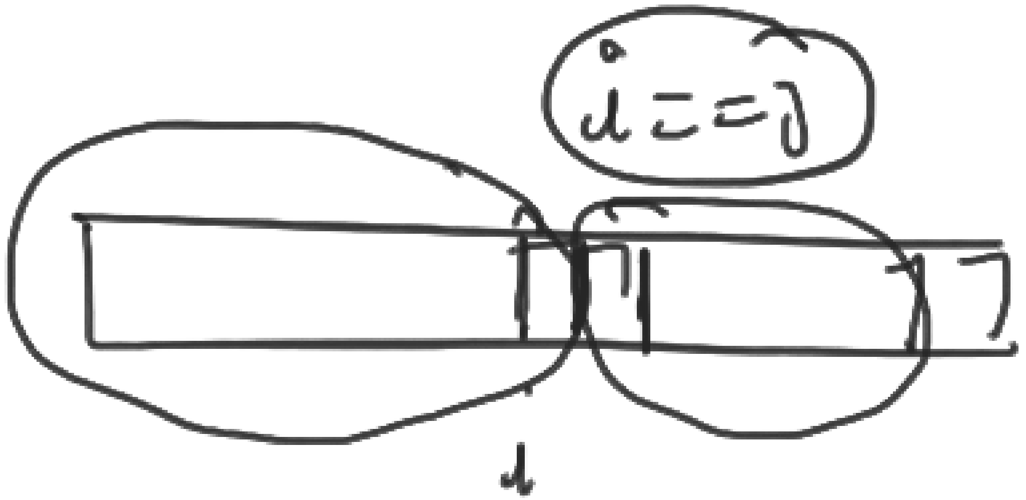
Hoare Partition.



A[i] > Pivot

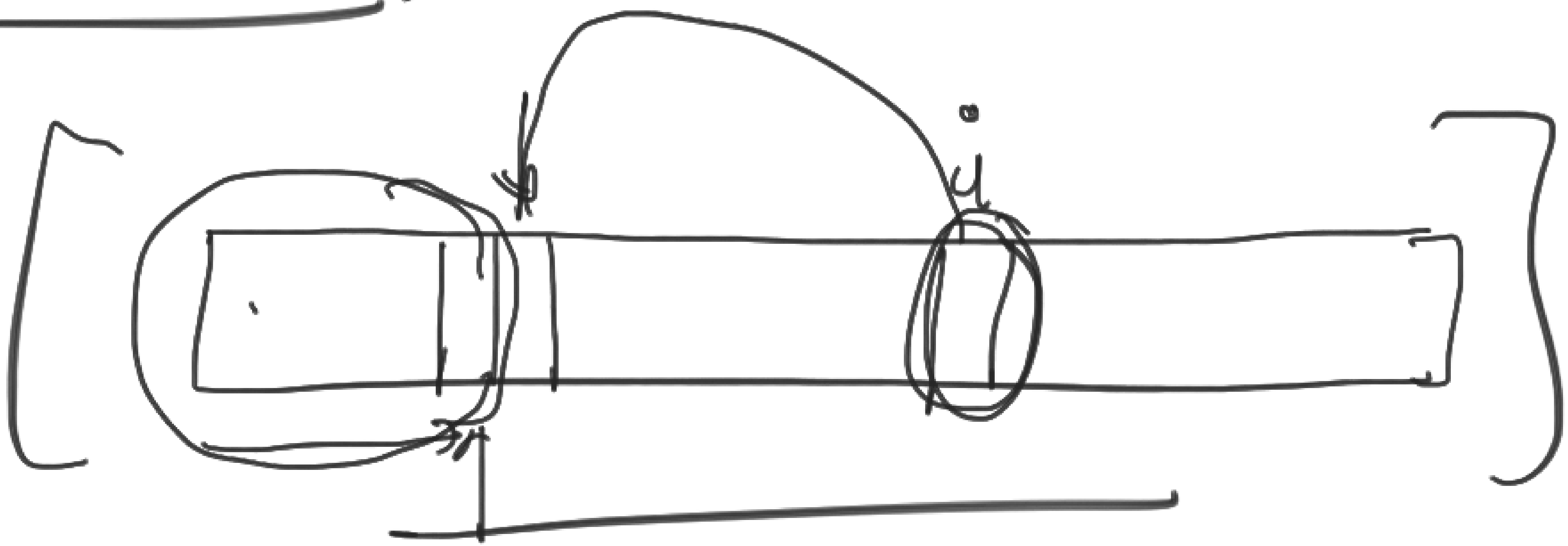
A[j] < pivot

$i > j$




```
while(i ≤ j)
{
    while(A[i] < pivot)
        i = i+1
    while(A[j] > pivot) ⇒ j = j-1
}
```

```
if (i <= j)
    swap(A[i], A[j])
}
```



→ move zeroes

→ Sort 0, 1 and 2

O(n log n)

2	1	0	1	0	2	1	0

O(n)

0	0	0	1	1	1	2	2

→ Remove duplicates.

stability

1	0	1	0	1	0	1

1	1	1	1	0	0	0

+ve, -ve

+ve	−

-ve +ve

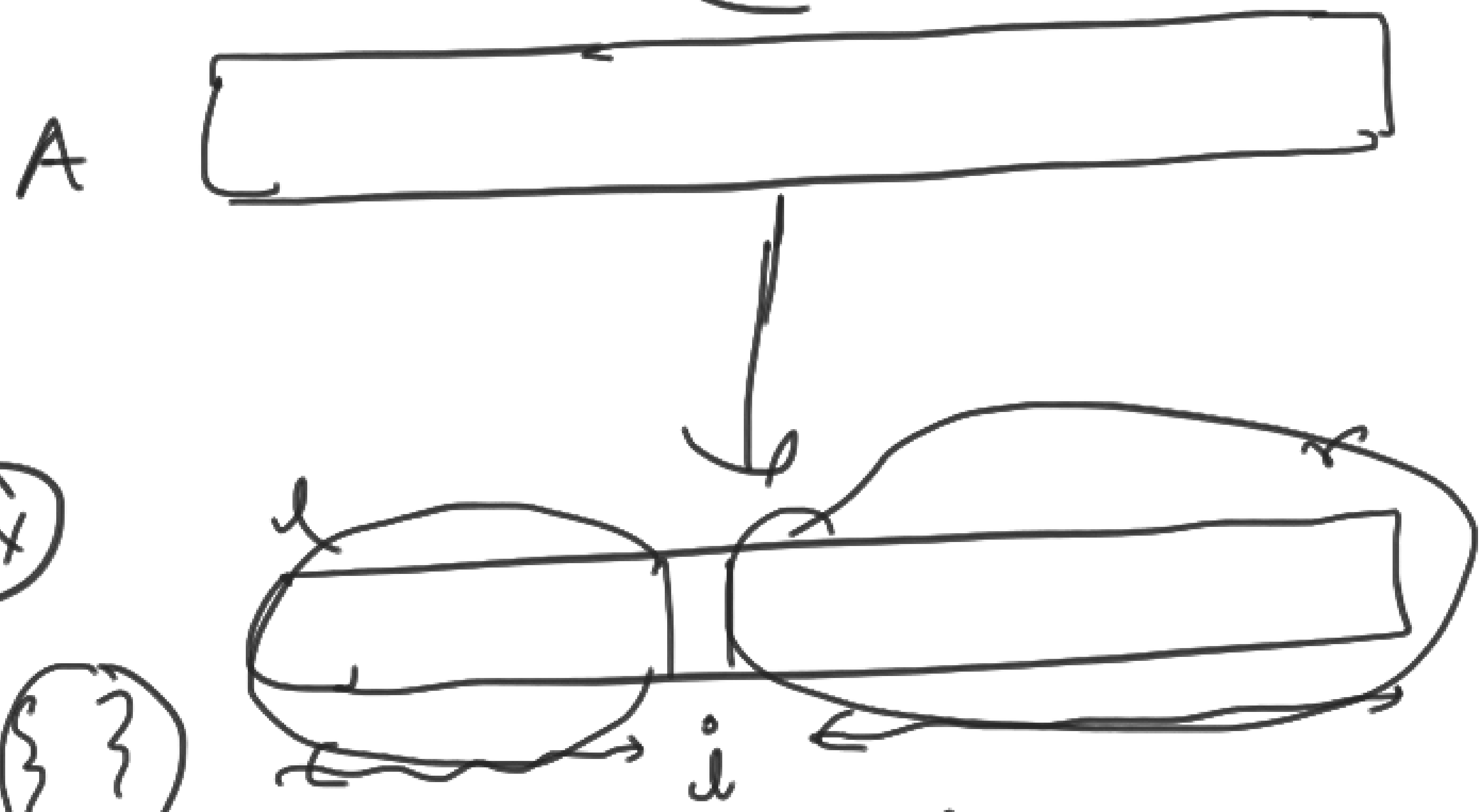
Time complexity

Base case

(l, r)

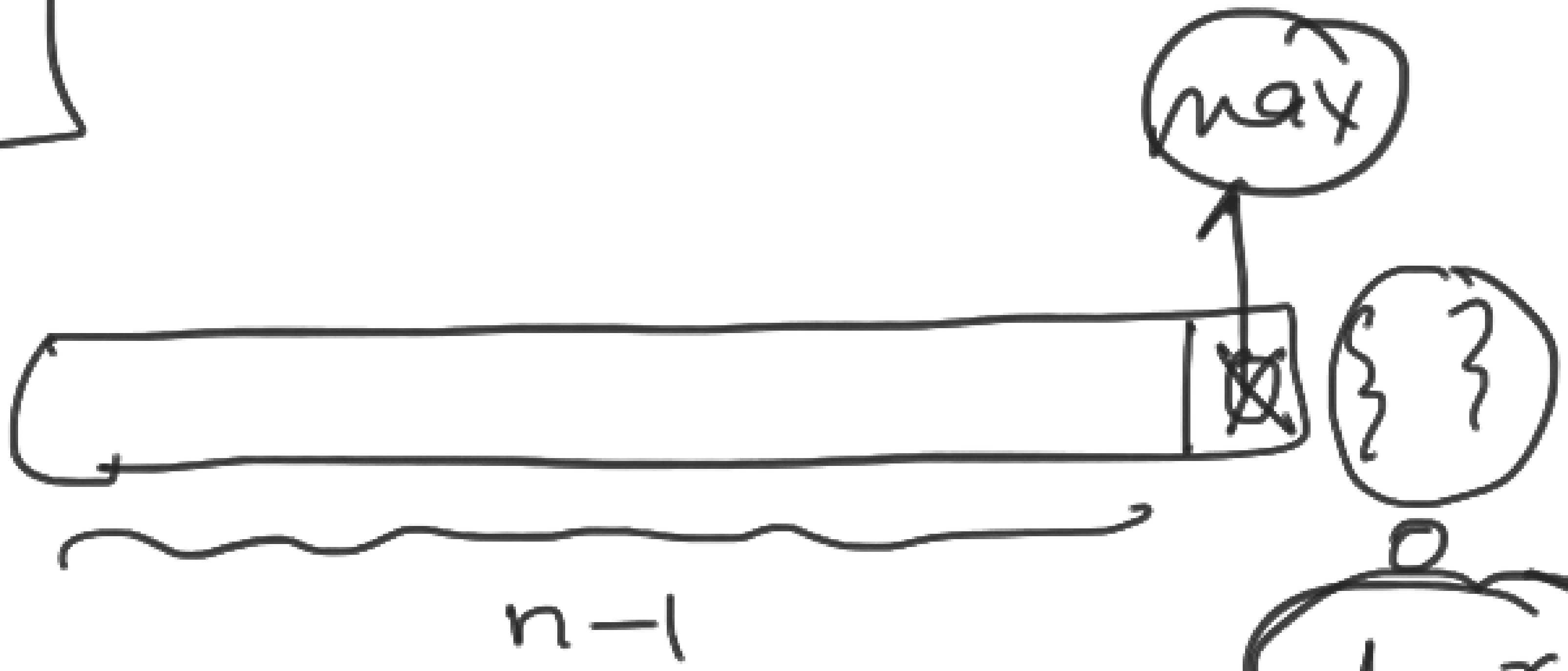
max

$(l == r)$ 2.



$(l, i-1)$ $(i+1, r)$

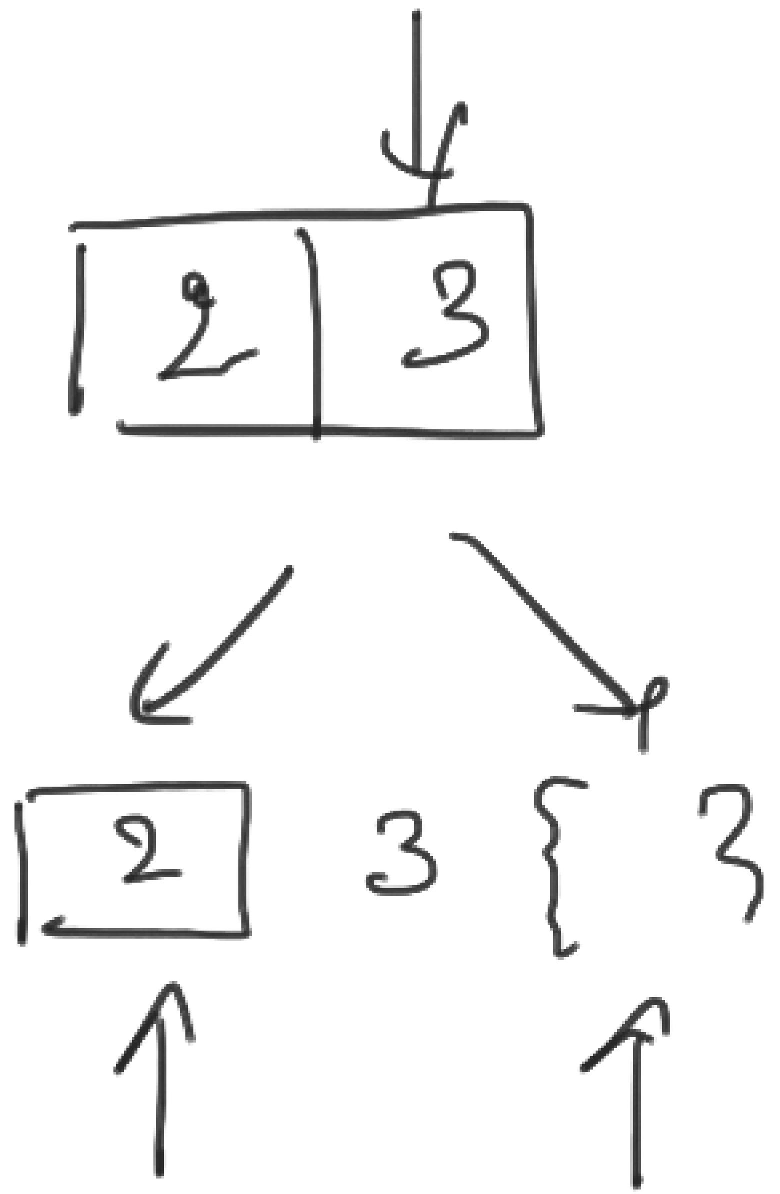
3

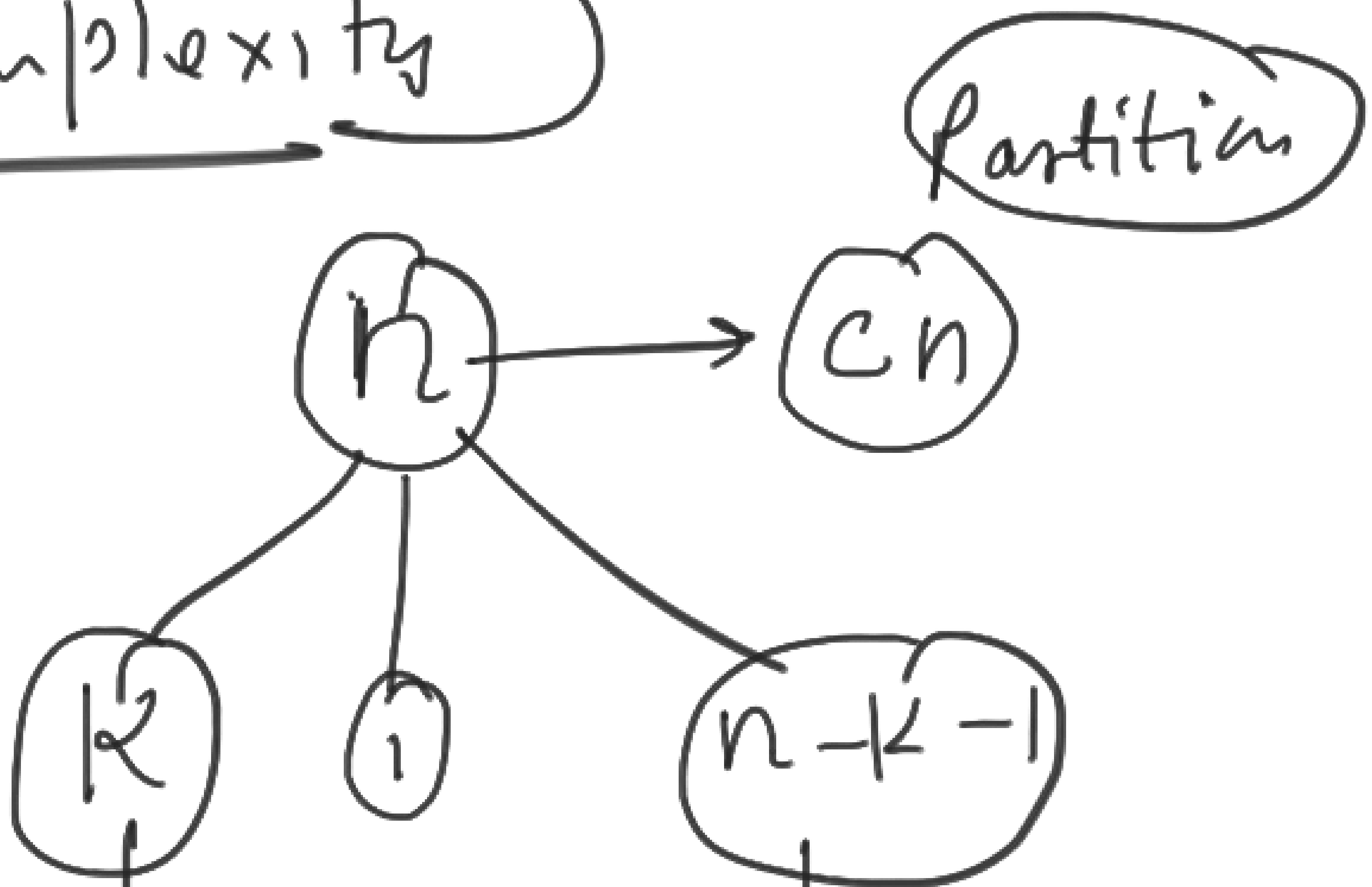


$l > r$

$l \geq r$ Base case $l > r$ $l == r$

Time complexity





$$\boxed{T(n) = T(k) + T(n-k-1) + cn}$$

choice of pivot

(0 , n-1)

(1 , n-2)

(2 , n-3)

|

(n/2 , n/2..)

↓

(n-2, 1)

(n-1 , 0)

unbalanced Partition (worst case)

Best case

1	2	3	4	5
1	2	3	4	5
1	2	3	4	0

0

5 4 3 2 1

$T(n) = 2T(n/2) + cn$

$O(n \log n)$

$$T(n) = T(0) + T(n-1) + cn$$

$$\boxed{T(n) = T(n-1) + cn}$$

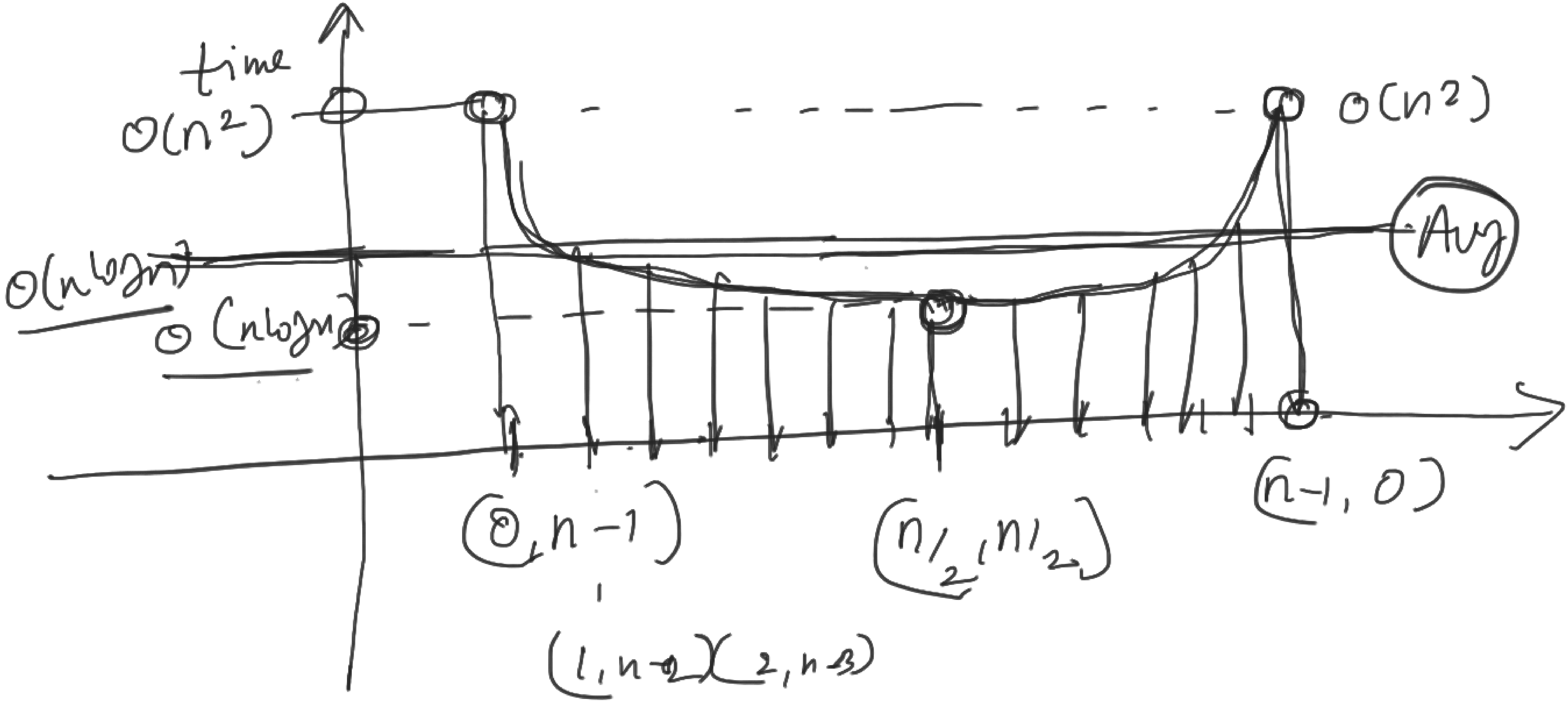
worst case

$$\begin{tikzcd}
n \arrow[r] \arrow[d] & cn \\
n-1 \arrow[r] \arrow[d] & c(n-1) \\
n-2 \arrow[r] \arrow[d, dash] & c(n-2) \\
1 & c
\end{tikzcd}$$

$$cn + c(n-1) + c(n-2) \; - - - \; c$$

$$c\left[n + n-1 + - - - 1\right]$$

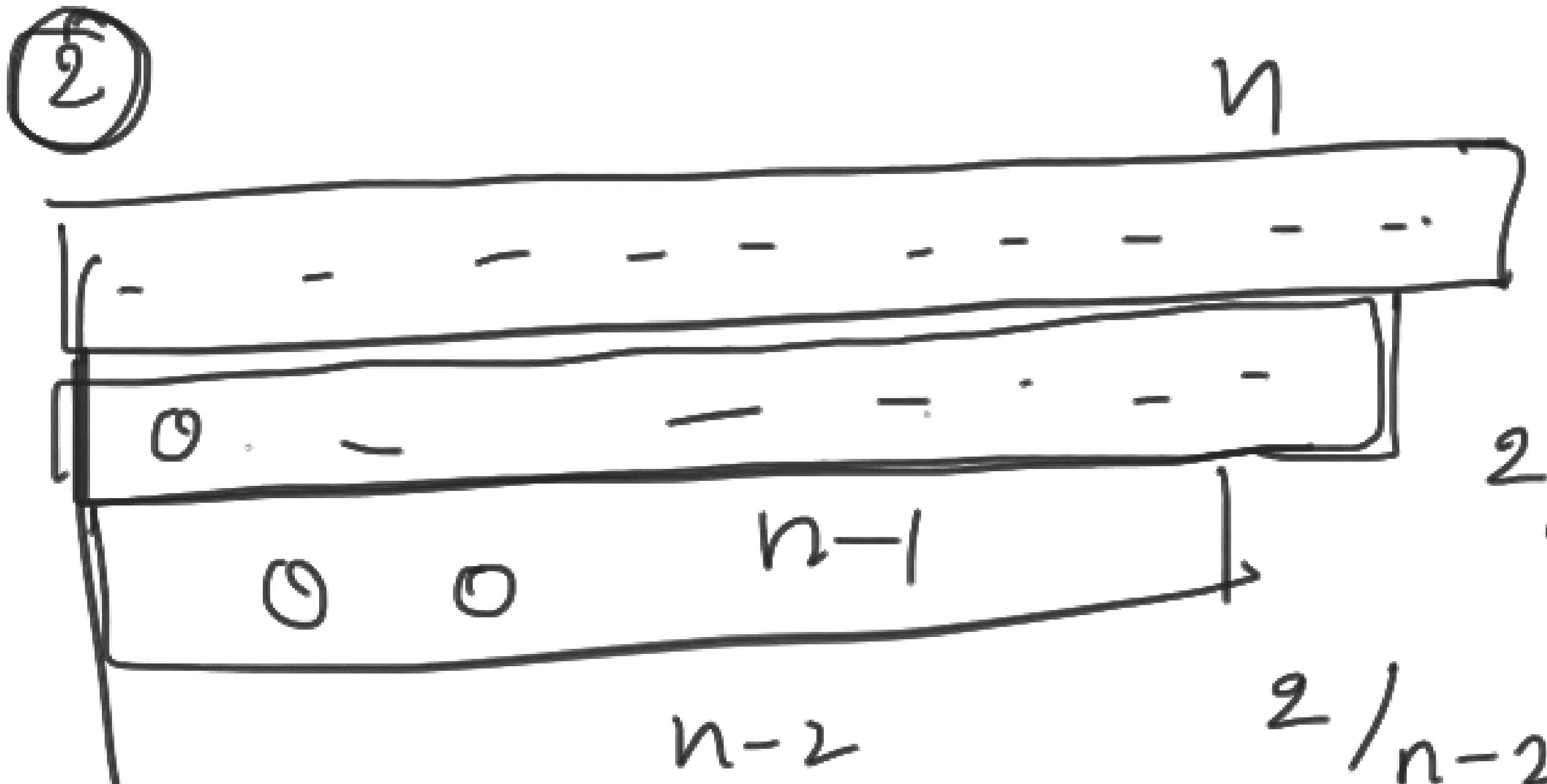
$$\frac{cn(n+1)}{2} = O(n^2)$$



time
O(n²)
O(n logn)
O (nlogn)
O(n²)
Avg
(0, n-1)
(1, n-2) (2, n-3)
(n/2, n/2)
(n-1, 0)

randomly

pivot



$2/n$

$2/n-1$

$2/n-2$

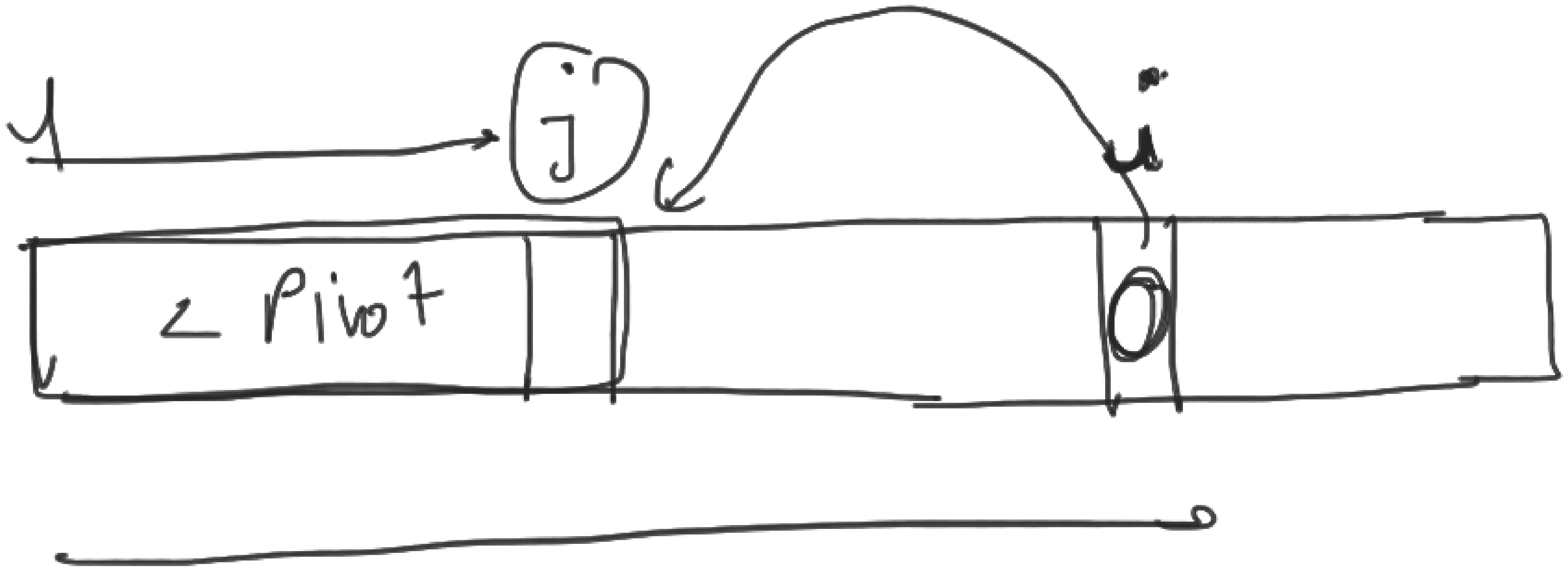
$\left(\frac{2}{n}\right)^n$

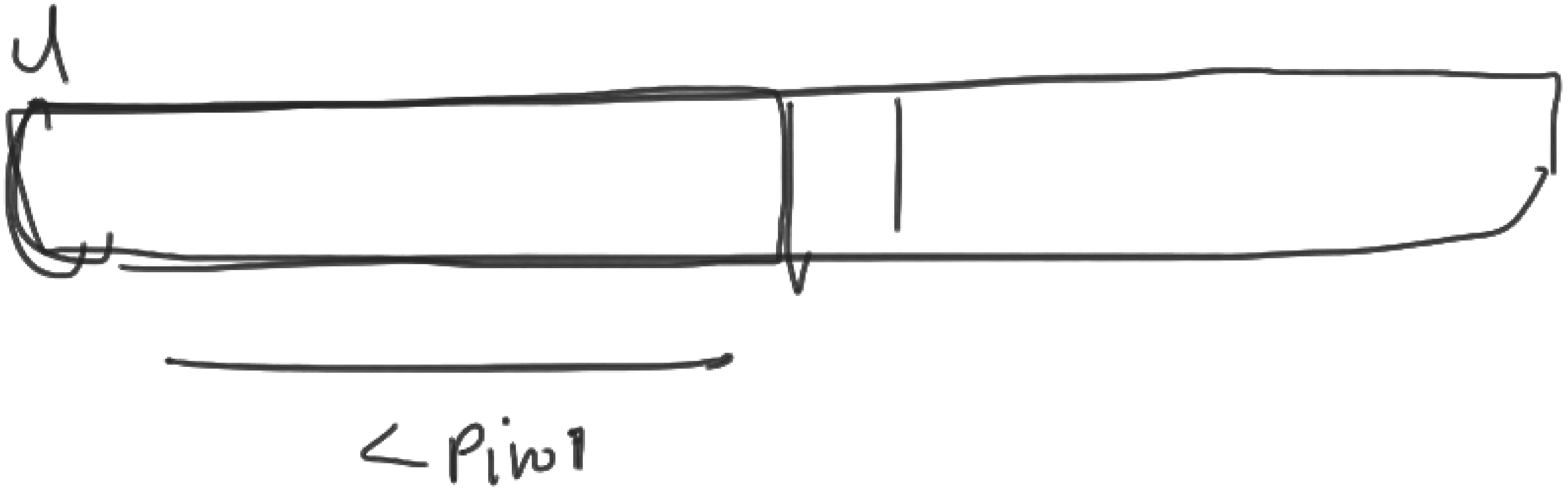
$$\left(\frac{2}{n} \times \frac{2}{n-1} \times \frac{2}{n-2} \cdots \frac{2}{1}\right) = \frac{2^n}{\lfloor n} \sim \frac{2^n}{n^n}$$

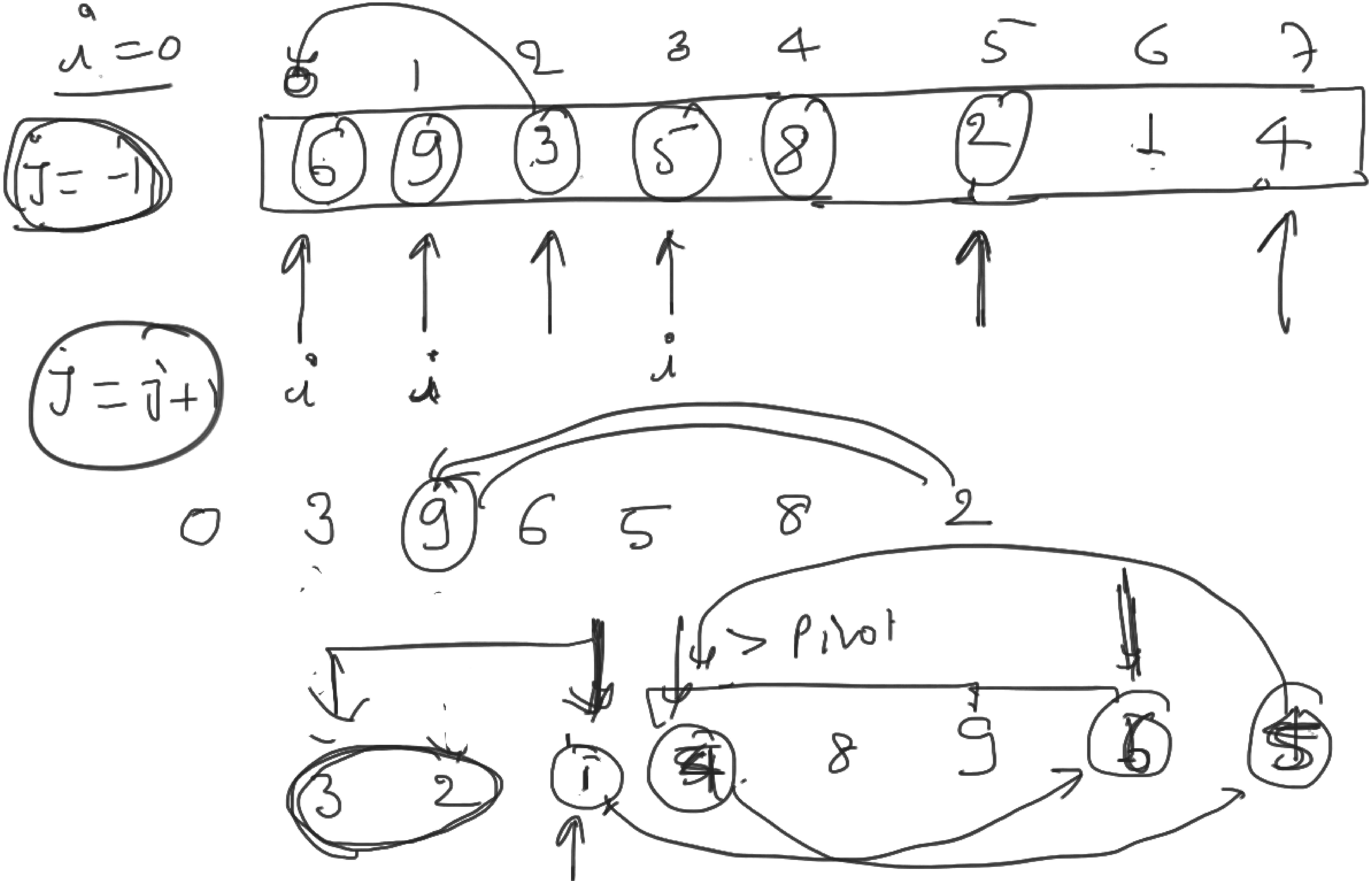
n times

$n^n \sim$

$$= \left(\frac{2}{n}\right)^n$$


l
j
i
< Pivot
0


l
< Pivot

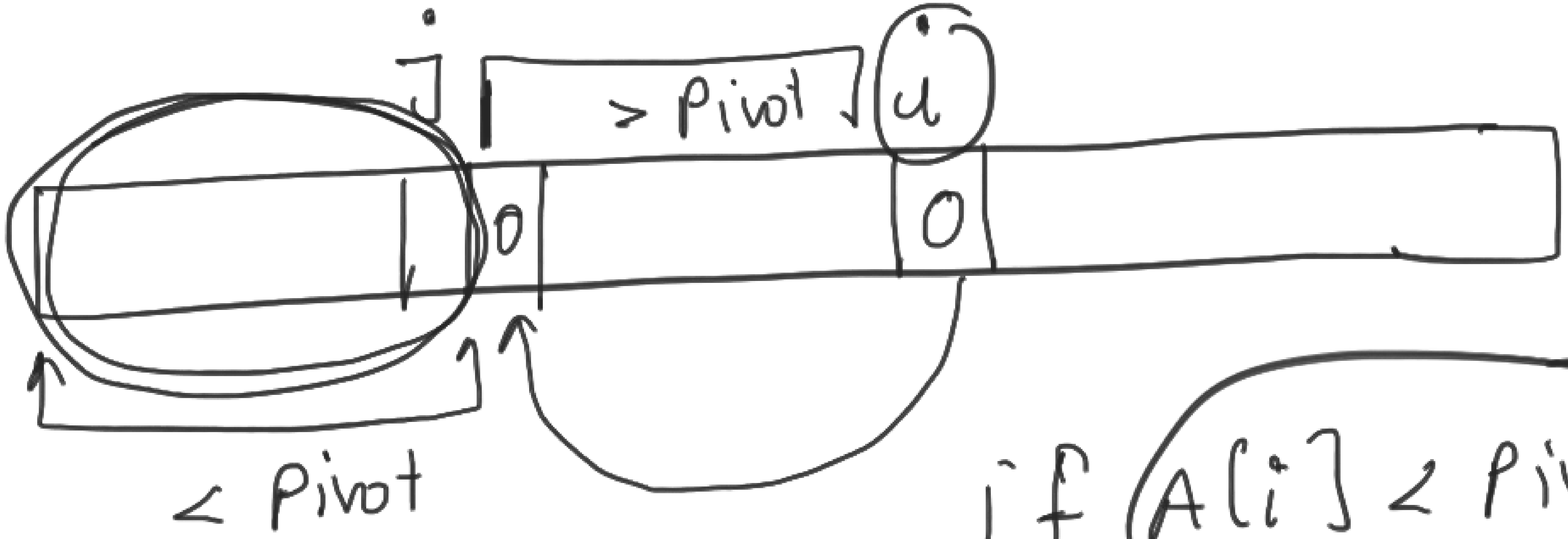


i = 0
0 1 2 3 4 5 6 7
J = -1
6 9 3 5 8 2 1 4
J = J+1
i i i
0 3 9 6 5 8 2
4 > Pivot
3 2 1 4 8 9 6 5

$j = l-1$

$j = l$

$\Uparrow$



if $A[i] < pivot$

$\{$ $j = j+1$

swap $(A[j], A[i])$

$j+1$

sort 0, 1, 2

TC