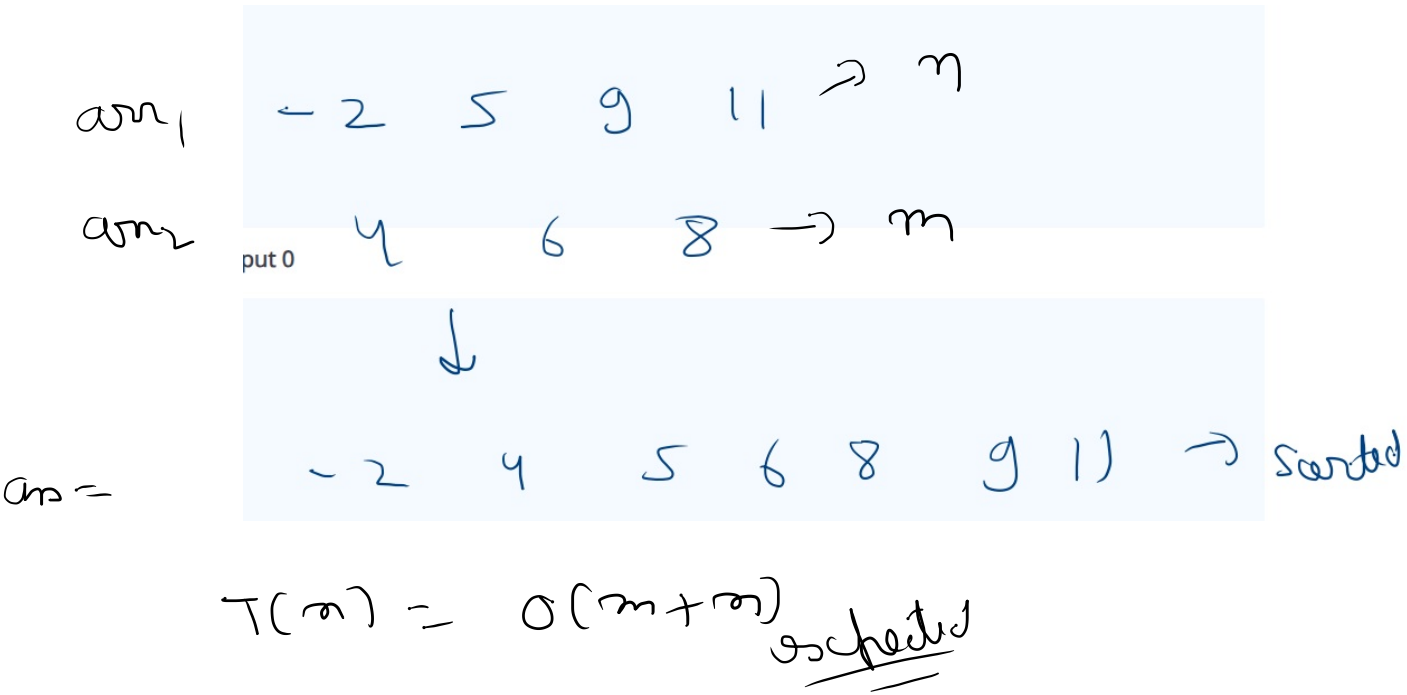
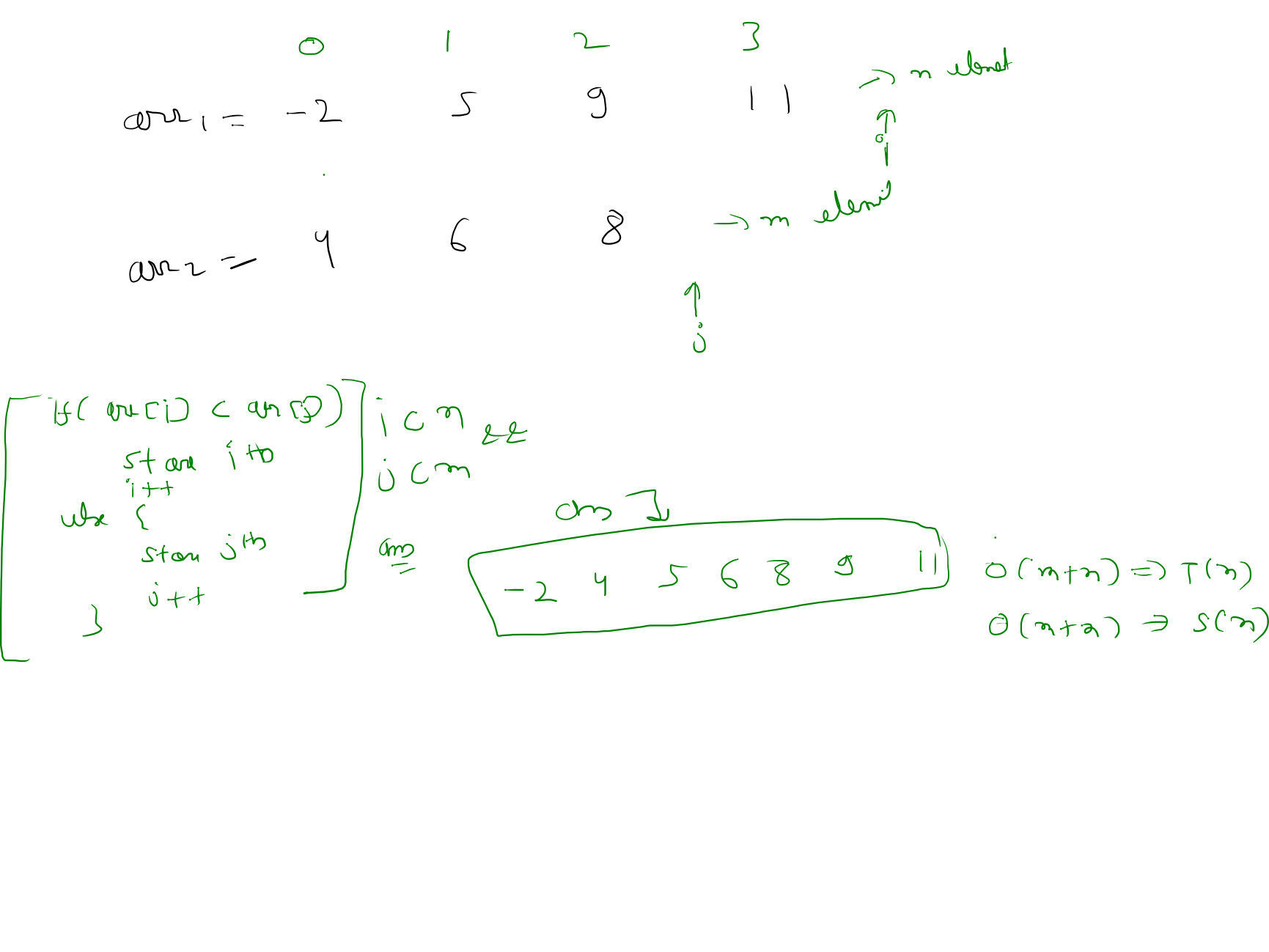
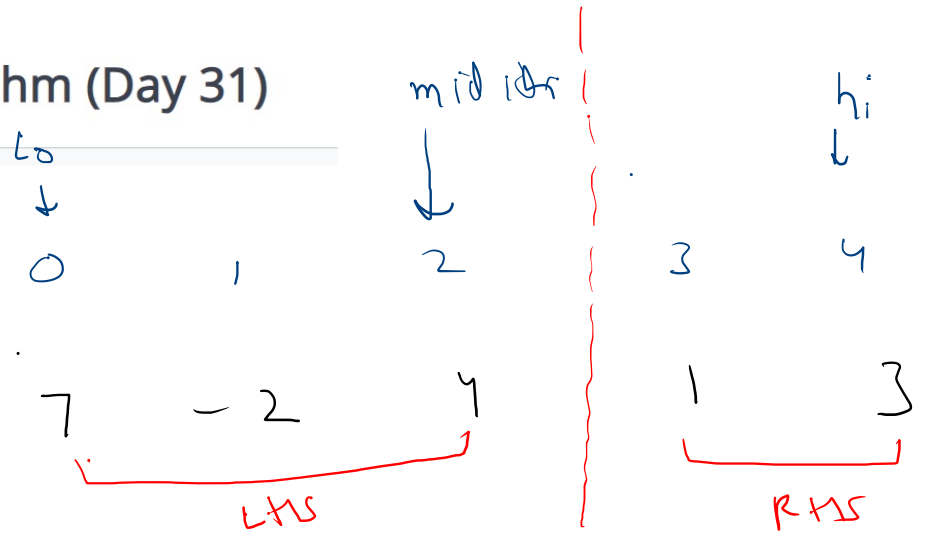
Merge Two Sorted Arrays (Day 31)



$T(n) = O(n+m)$ optimal


0 1 2 3
arr1 = -2 5 9 11
n elements
arr2 = 4 6 8
m elements
i
j
if(arr[i] < arr[j])
store i th
i++
else {
store j th
j++
}
i < m &&
j < m
arr
-2 4 5 6 8 9 11
O(n+m) ⇒ T(n)
O(n+n) ⇒ S(n)

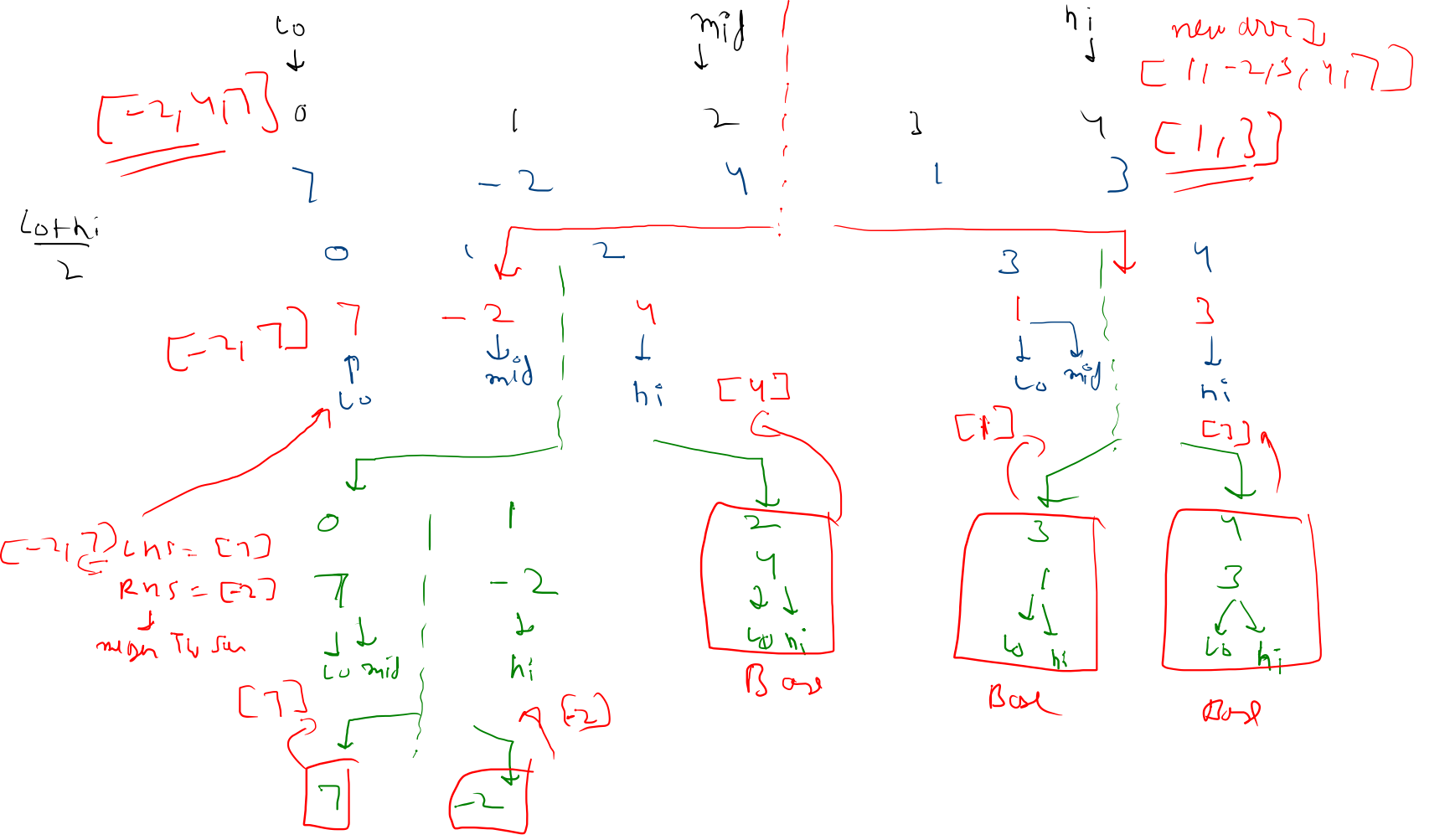
Merge Sort Algorithm (Day 31)



$$\text{mid idx} = \frac{\text{lo} + \text{hi}}{2} = \frac{0+4}{2} = \frac{4}{2} = 2$$

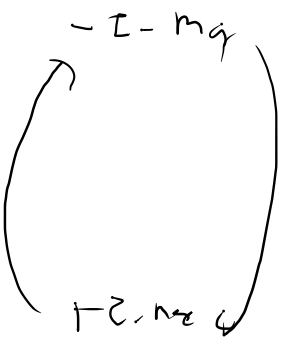
only Am 2

Faith ⇒ LHS = -2 4]] ⇒ merge Two Sorted arr
RHS = 1 3


lo
mid
hi
new arr
[-2,4,7]
[1,3]
lo+hi
2
[-2,7]
mid
[4]
[1]
[7]
[-2]
LHS = [7]
RHS = [-2]
Box
Box
Box

Cal of mid value

$$mid = \frac{lo + hi}{2}$$ ✗



$$mid = lo + \frac{(hi - lo)}{2}$$ ✓

Eg $lo = 1$ $\quad hi = 2 . mid$

$$mid = \frac{lo + hi}{2} = \frac{1 + 2 . mid}{2} = \frac{-2 . mid}{2}$$ ✗

$$mid = 1 + \frac{(2 . mid - 1)}{2} =$$ ✓

$lo + \frac{hi - lo}{2}$

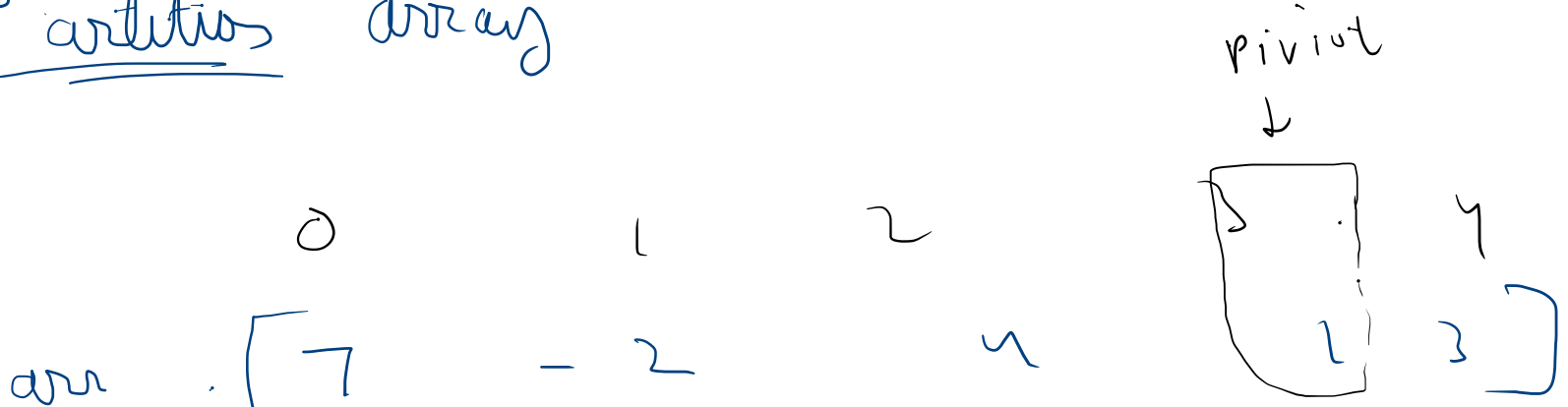
$\frac{2lo + hi - lo}{2}$

$\frac{2lo - lo + hi}{2} \Rightarrow \frac{lo + hi}{2}$

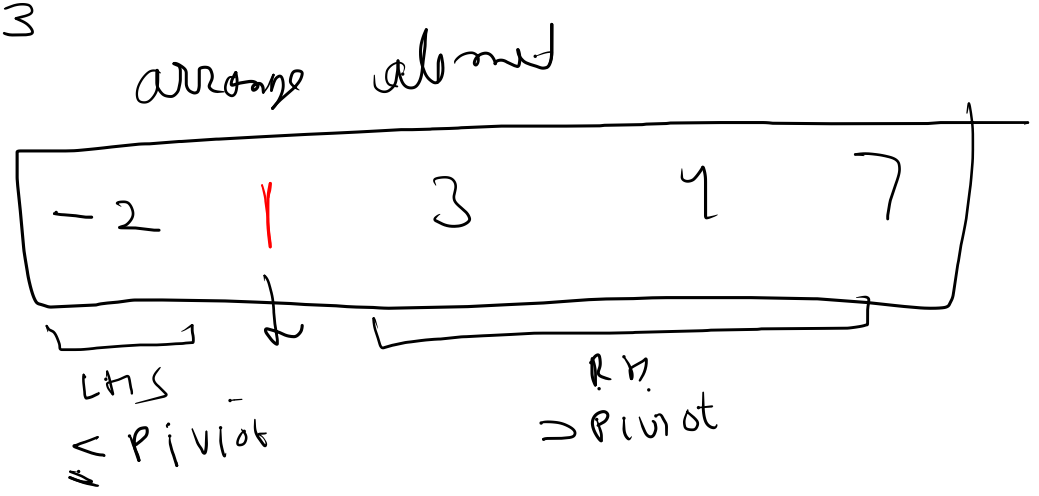
$T(n) = n \log n$

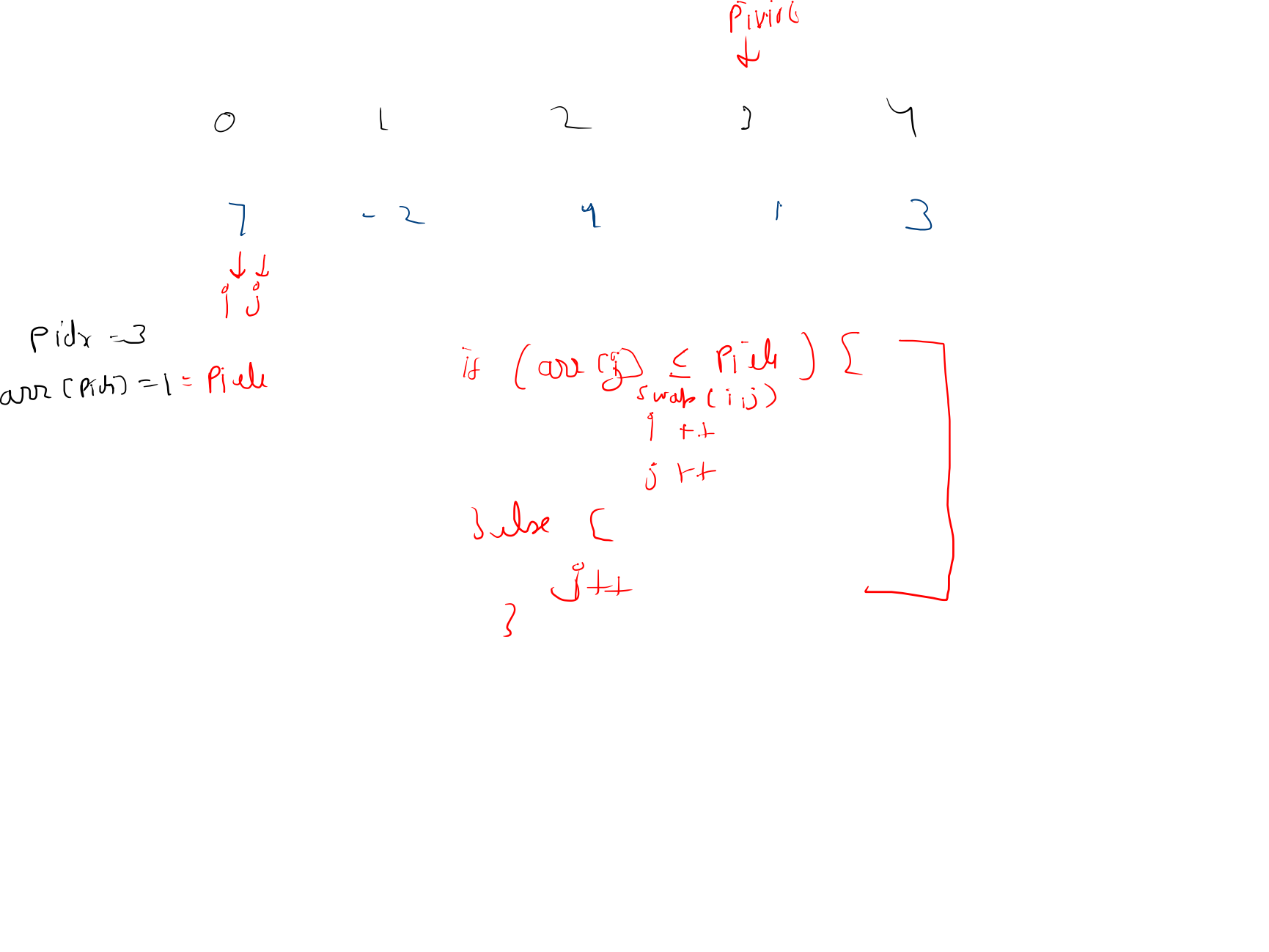
$S(n) = n$

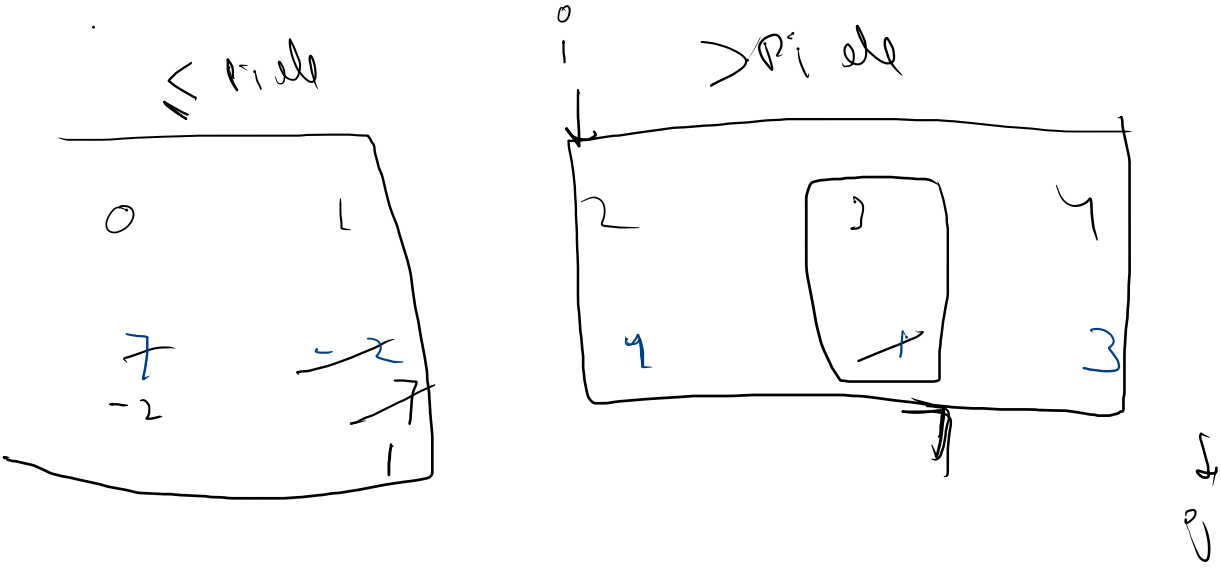
Partition array



pi = 3




Pivot
0 1 2 3 4
7 -2 4 1 3
i j
pidx = 3
arr[pidx] = 1 = pivot
if (arr[j] ≤ pivot) {
swap(i, j)
i++
j++
} else {
j++
}



① $0 \ominus i-1$ → ≤ Pivot

② $i \ominus j-1$ → > Pivot

③ $j - n-1$ → Unprocessed

```
if ( arr[j] ≤ Pivot ) {
    swap (i, j)
    i++
    j++
} else {
    j++
}
```

1 ≤ 1

3 ≤ 4

i j
↓ ↓

-2 ~~7~~ ~~1~~ 3
1 7

↓
j

$-2 \leq 1$

$7 \leq 1$

$1 \leq 1$

$3 \leq 1$

Quick Sort

lo				hi
0	1	2	3	4
7	-2	4	1	3

pidx = hi

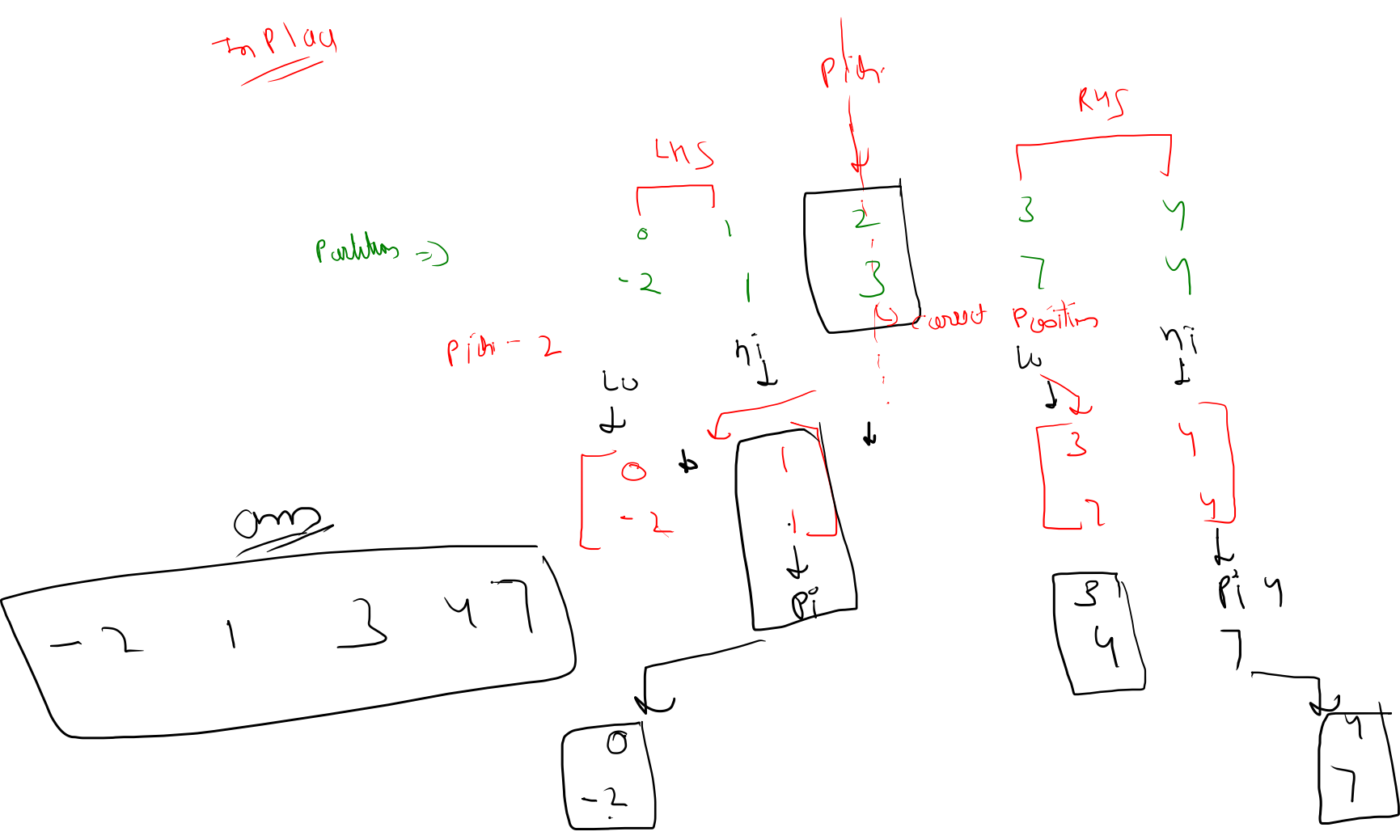
arr[pidx] = 3] ⇒ Partition arr

pidx				
0	1	2	3	4
-2	1	3	7	4

Partition ⇒

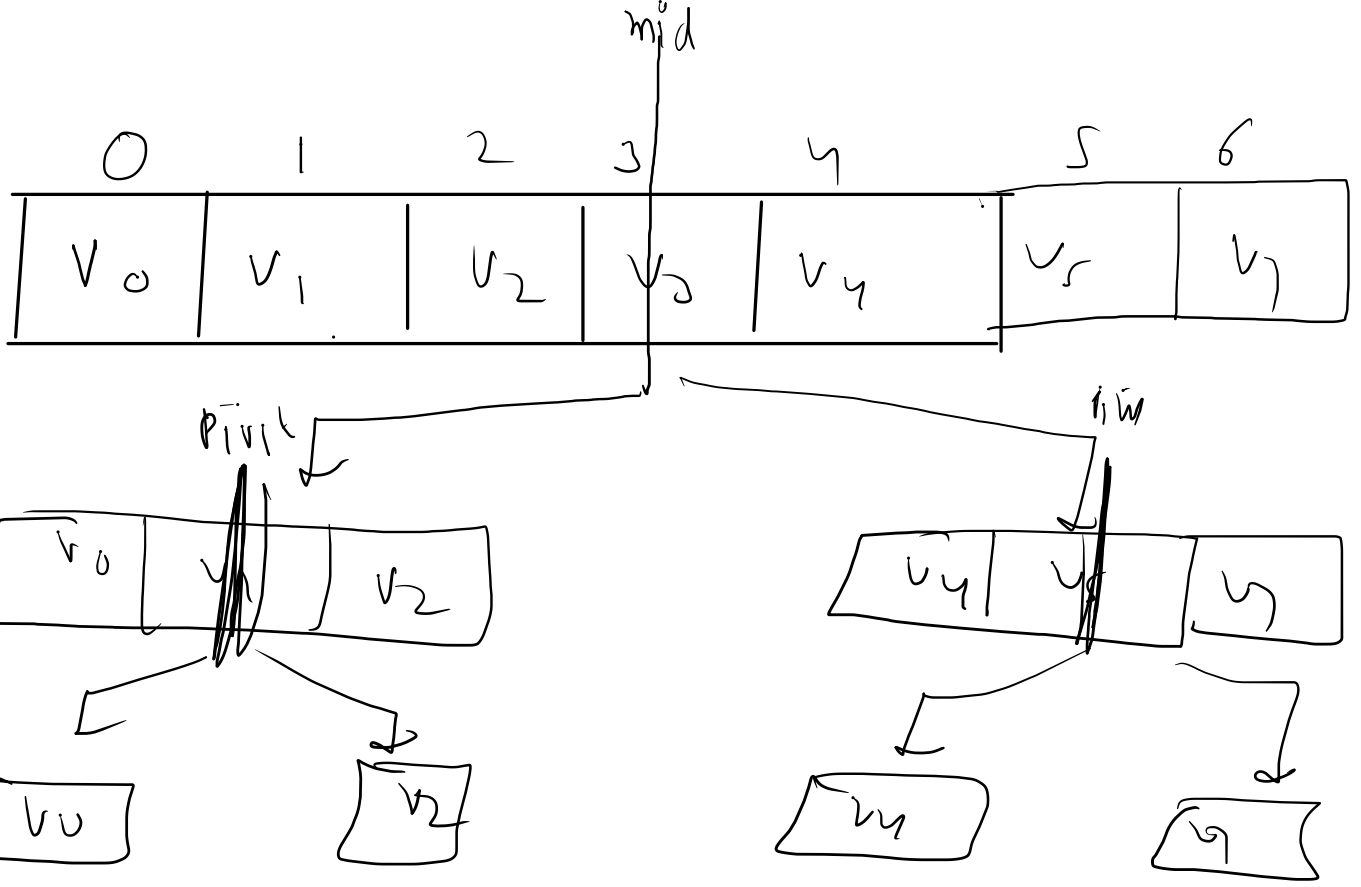
↳ correct Position

pidx = 2


In Play
Pivot
LHS
RHS
Partition =>
0
1
-2
1
2
3
3
4
7
4
Pivot - 2
Lo
Hi
Correct Position
Lo
Hi
0
-2
1
1
Pi
3
7
4
4
Pi
4
7
-2 1 3 4 7

Best
Case

Time



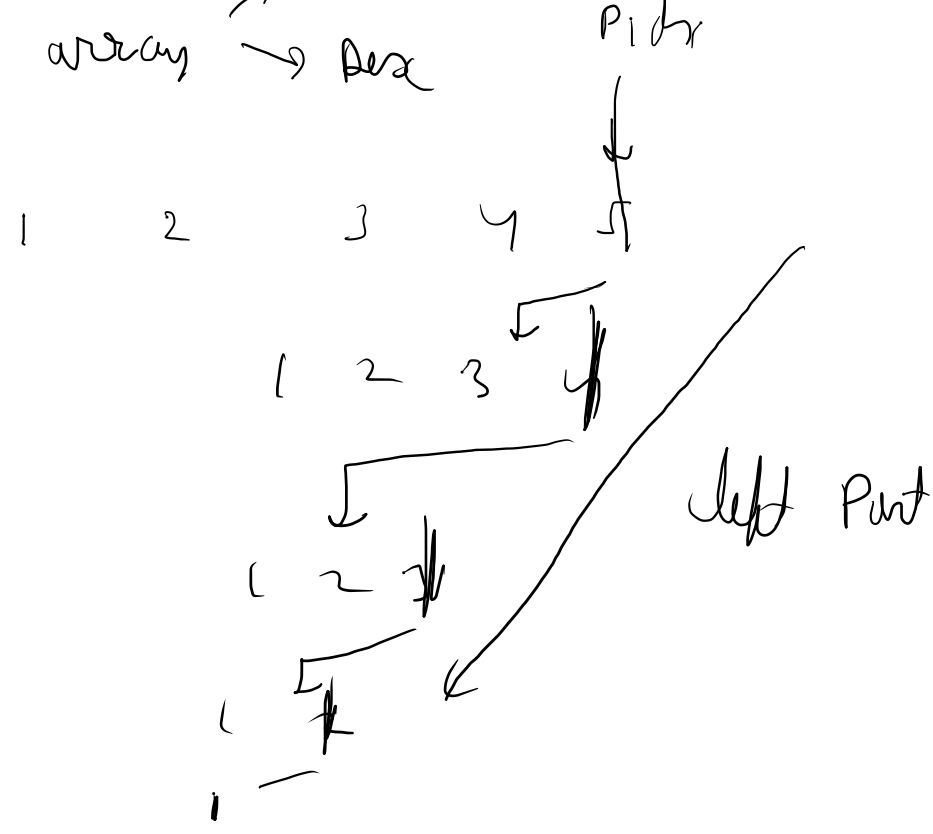
Worst TC = $O(n^2)$

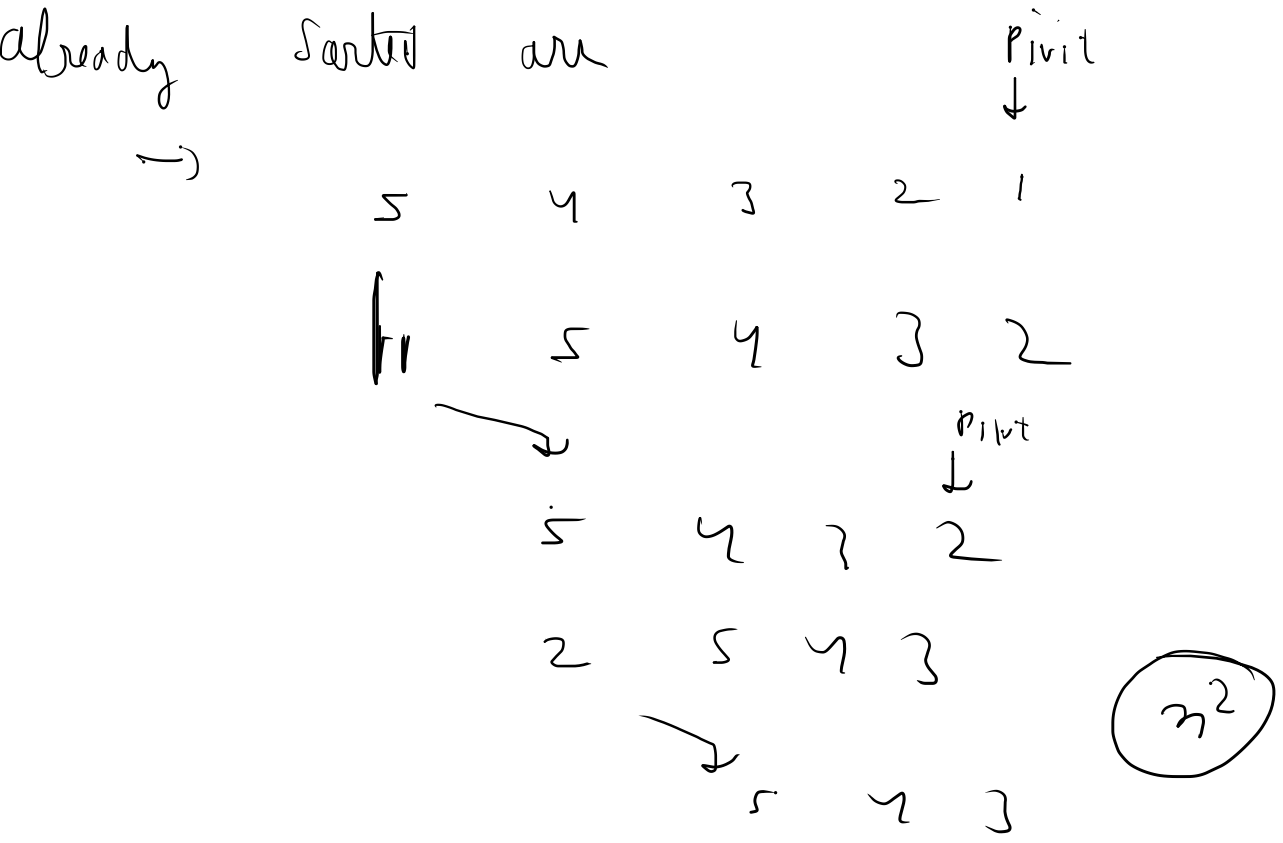
n^2

=> Sorted array → Asc

→ Desc

already




already sorted arr
Pivit
5 4 3 2 1
5 4 3 2
Pivot
5 4 3 2
2 5 4 3
5 4 3
n^2