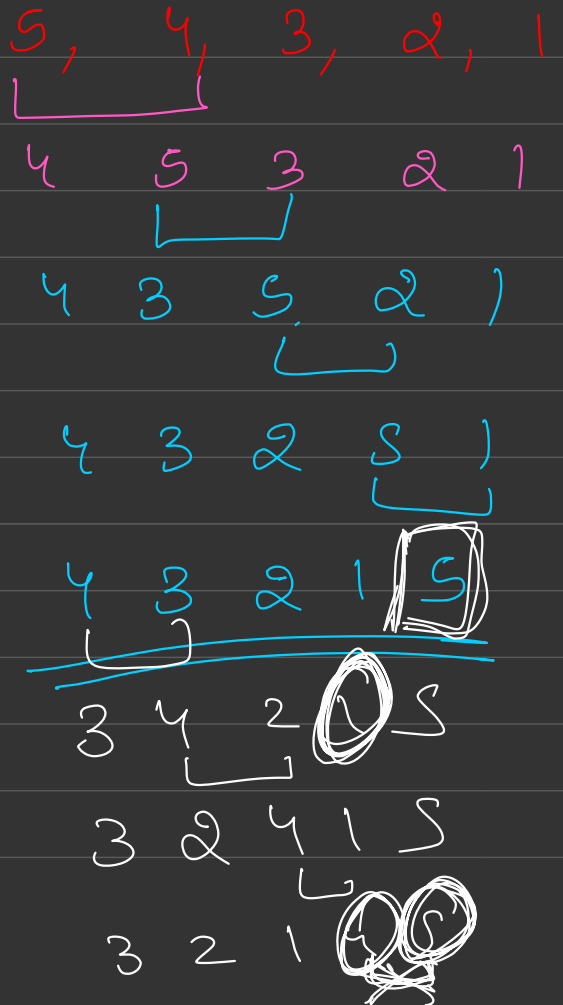
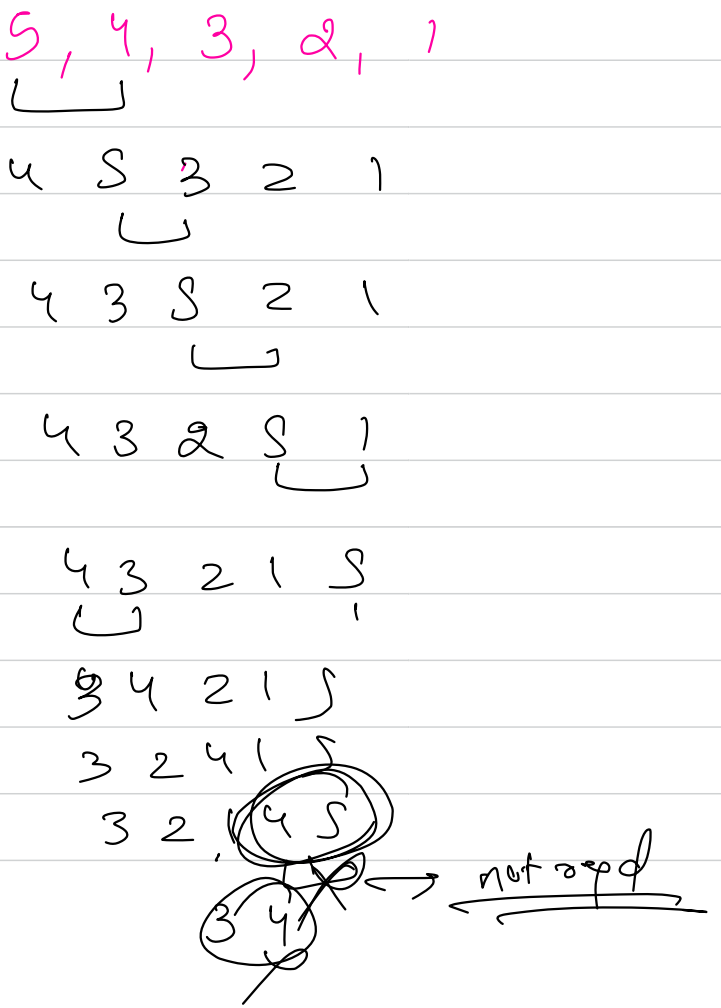
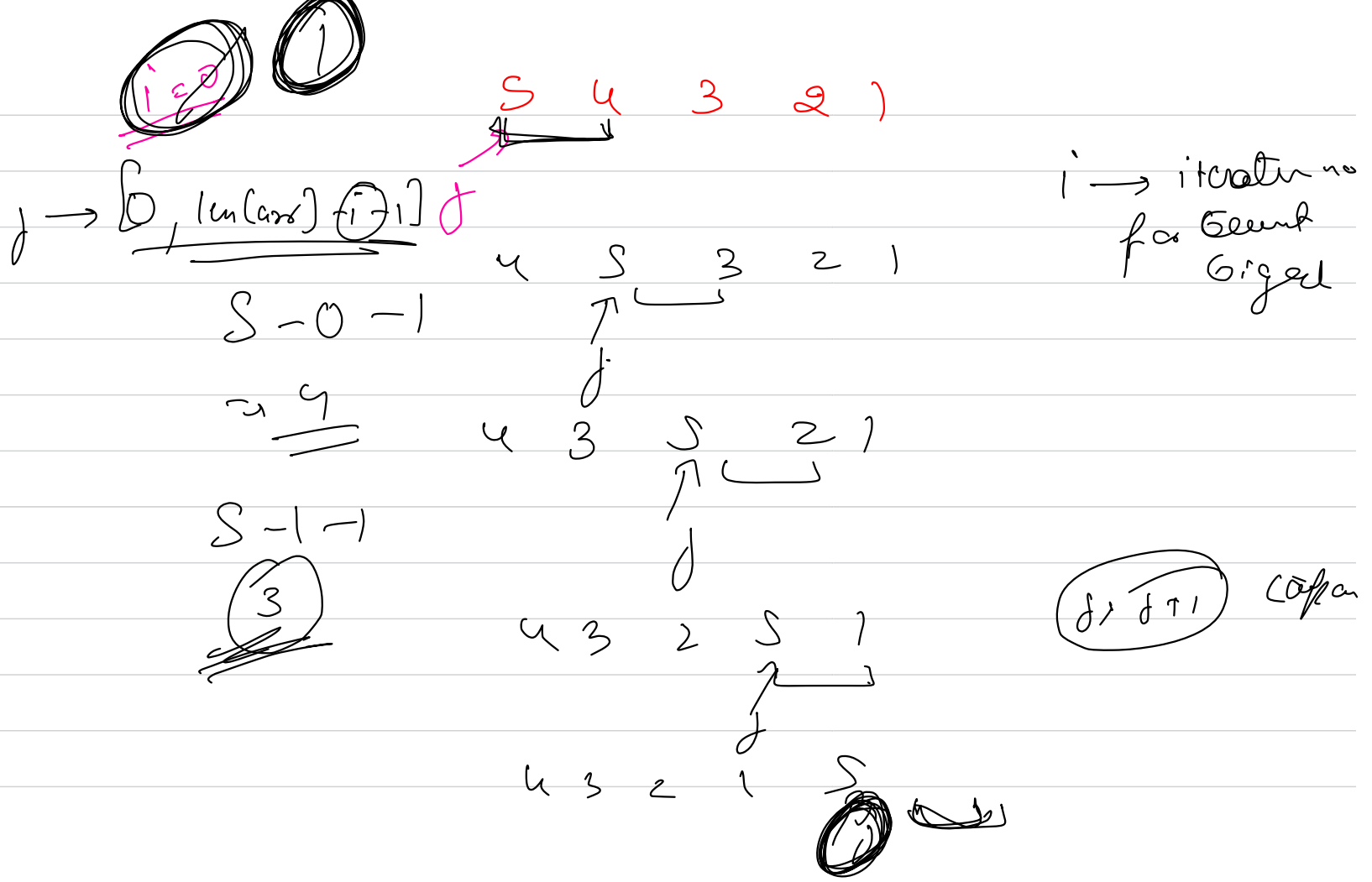
Agenda → Code Bubble Sort

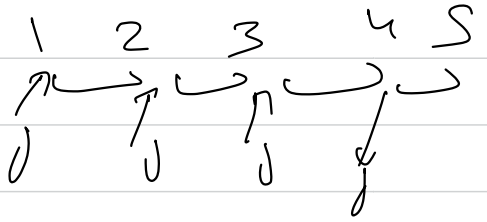
↳ Insertion sort

↳ Counting Sort


5, 4, 3, 2, 1
4 5 3 2 1
4 3 5 2 1
4 3 2 5 1
4 3 2 1 5
3 4 2 1 5
3 2 4 1 5
3 2 1 4 5


5, 4, 3, 2, 1
4 5 3 2 1
4 3 5 2 1
4 3 2 5 1
4 3 2 1 5
3 4 2 1 5
3 2 4 1 5
3 2 4 5
3 4
not sorted


i=0
1
5 4 3 2 1
j → [0, len(arr) - i - 1] j
i → iteration no for Gearup Gigel
5 - 0 - 1
→ 4
4 5 3 2 1
j
5 - 1 - 1
3
4 3 5 2 1
j
j, j+1 capa
4 3 2 5 1
j
4 3 2 1 5



5 4 2 3 4

1 5 2 3 4

1 2 5 3 4

1 2 3 5 4

1 2 3 4 5

→ No swap will take place

```python
def bubble_sort(arr):
    for i in range(0, len(arr)):
        swapped = False
        for j in range(0, len(arr) - i - 1):
            if arr[j] > arr[j+1]:
                arr[j], arr[j+1] = arr[j+1], arr[j]
                swapped = True

        if not swapped:
            return # why ?? because now the list will be
    return
```

len(arr)-1

[5, 4, 3]

→ i → start from 0 & ends at len(arr)-1

i → [0, 4]

when i=0 → j [0, 3]

i	j	swapped
0	0	True
	1	True
	2	True
	3	True
1	0	True
	1	True
	2	True
2	0	False
	1	

0 1 2 3 4

1 2 3 4 5

when i=1 → j [0, 2]

i=2 → j [0, 1]

1 2 3 4 5

5 4 3
4 5

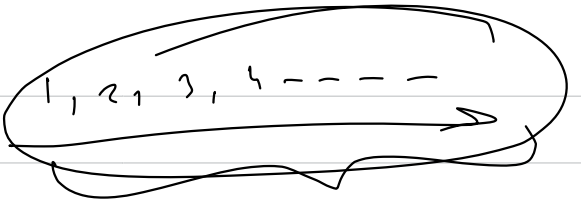
4 3 5
5 4

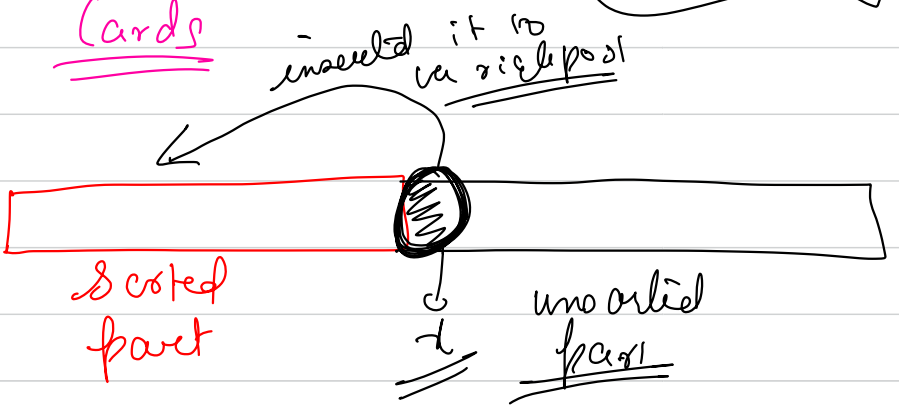
$n-1 + n-1 + n-1 - - - - - - n-1$

$a(n-1)$

$n-1 + n-2 + n-3 - - - - 1$

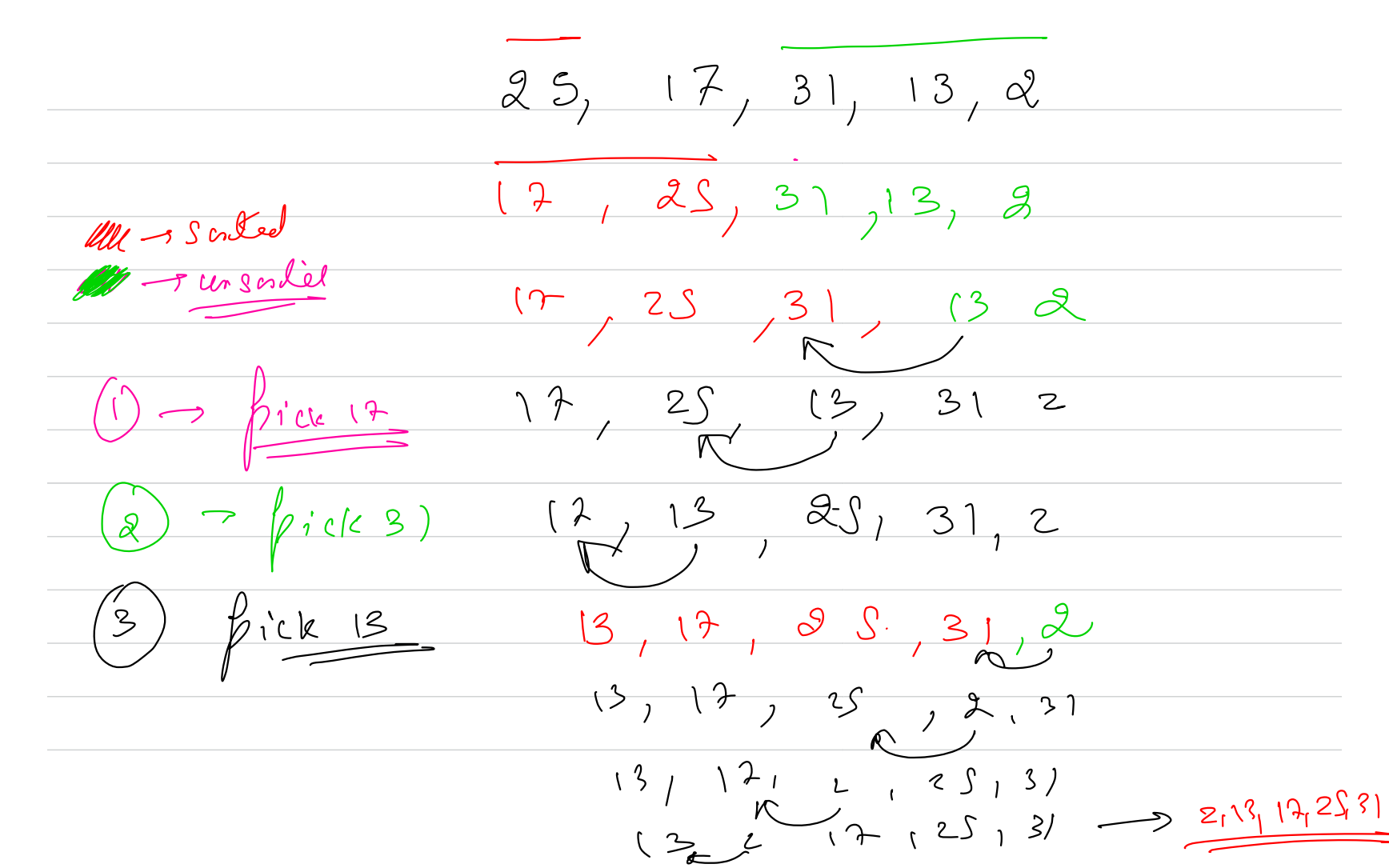
Insertion Sort :

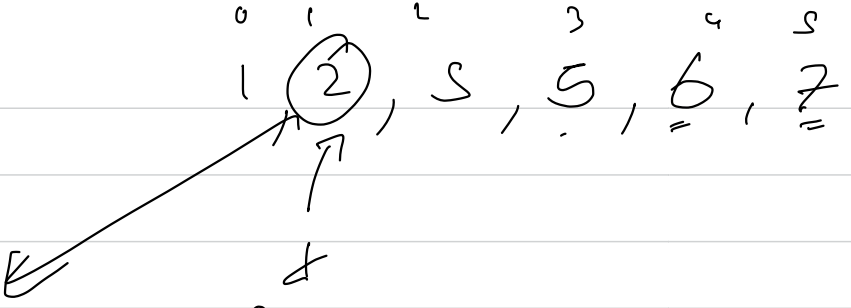




→ search first for the right position of x^{th} card

→ Insert that x^{th} card to the right pos.


25, 17, 31, 13, 2
17, 25, 31, 13, 2
sorted
unsorted
17, 25, 31, 13 2
17, 25, 13, 31 2
12, 13, 25, 31, 2
13, 17, 25, 31, 2
13, 17, 25, 2, 31
13, 17, 2, 25, 31
13, 2, 17, 25, 31
2, 13, 17, 25, 31
1 → pick 17
2 → pick 13
3 pick 13



i = 5

j = 4 3 2 1

key = 4

2 was the first elent less than the key

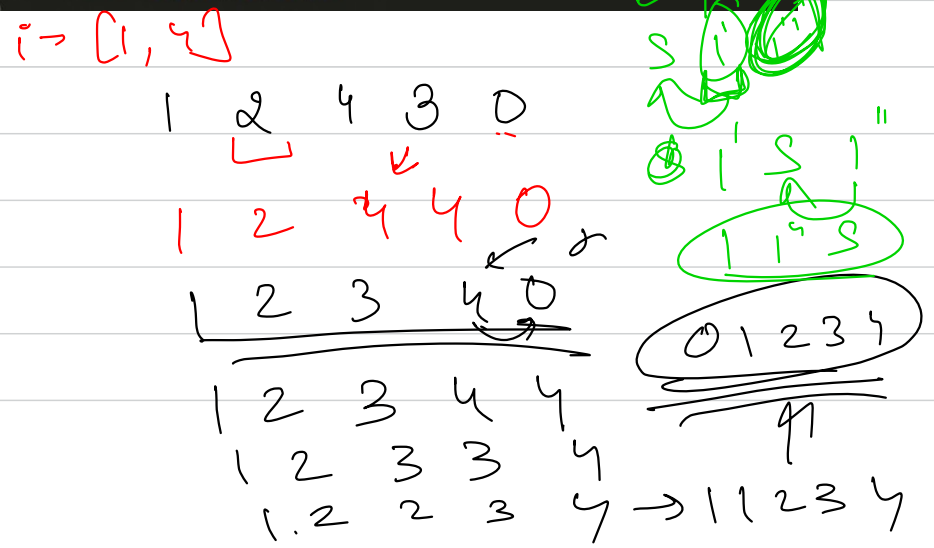
So right place of key will be to the next right of 2 (j+1)

```python
def insertion_sort(arr):
    for i in range(1, len(arr)):
        key = arr[i] # this element wi
        j = i - 1 # because the right
        while j >= 0 and key < arr[j]:
            arr[j+1] = arr[j]
            j -= 1
        arr[j+i] = key

    return

li = [5,4,3,2,1]
```

i	j	key
1	0	2
2	1	4
3	2	3
3	1	3
4	3	0
4	2	0
4	1	0
4	0	0
	-1	

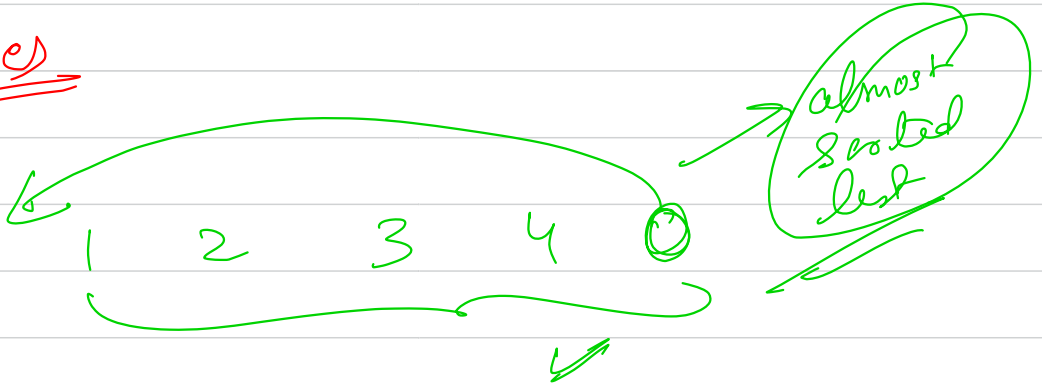


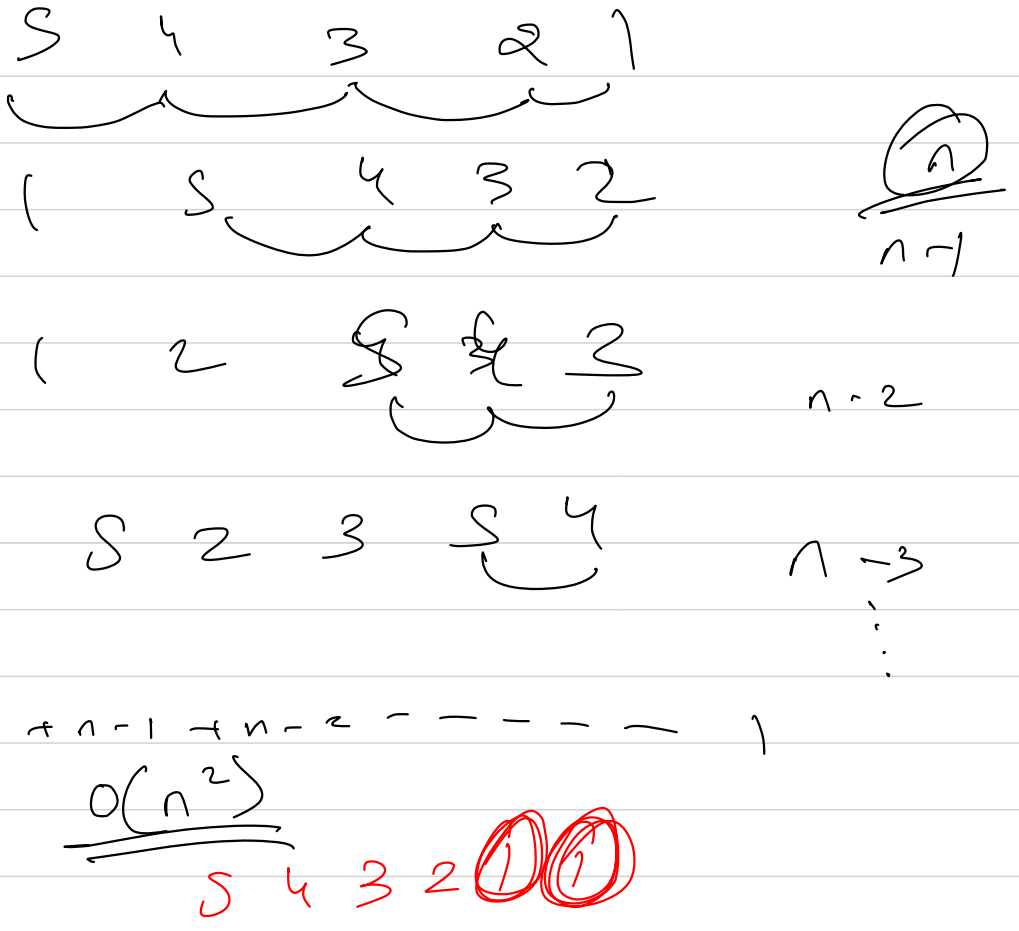
TC → $O(n^2)$ $\Theta(n^2)$, $\Omega(n)$ Space → $O(1)$

No of swaps → $O(n^2)$ In place → Yes

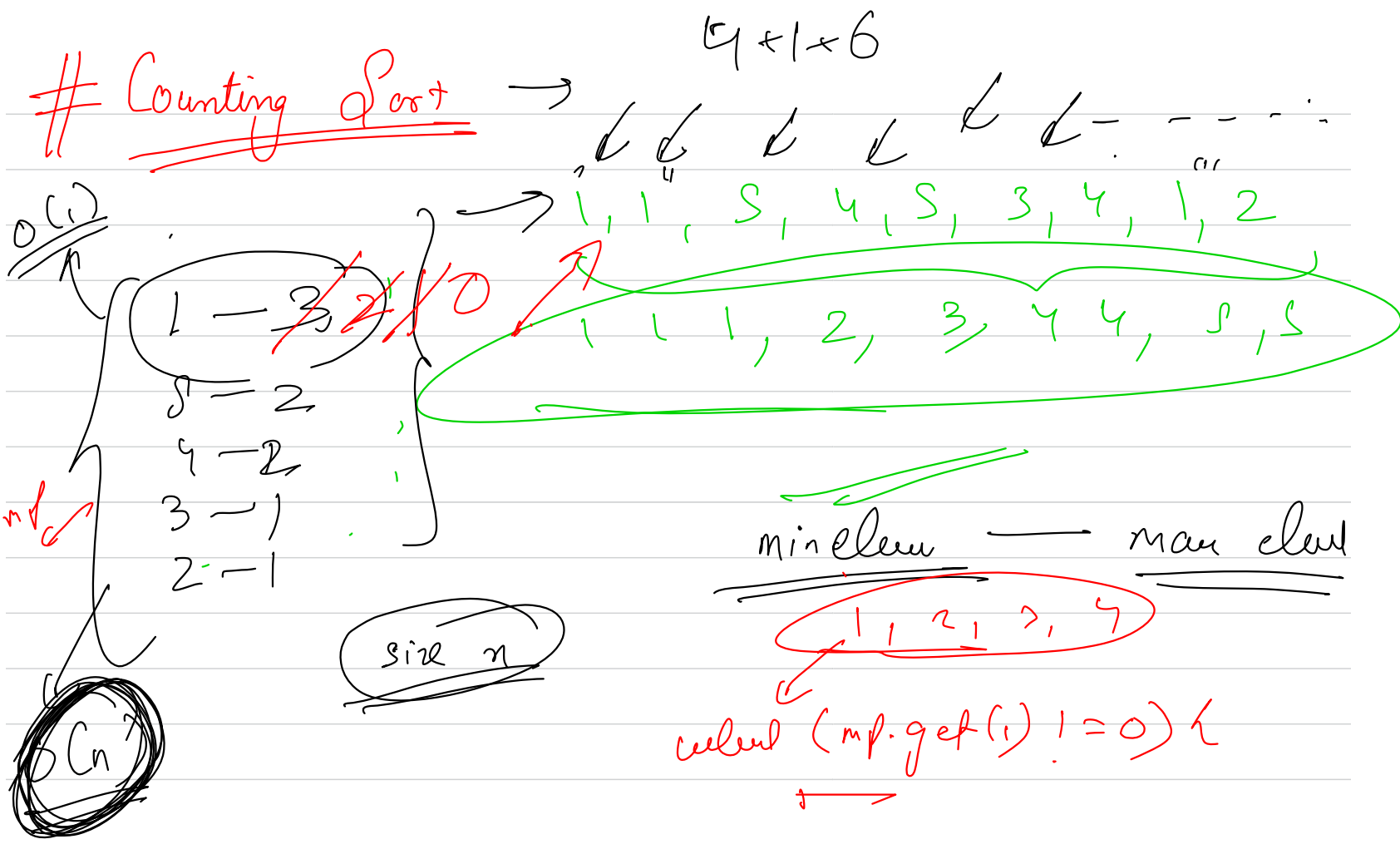
No of comparison → $O(n^2)$

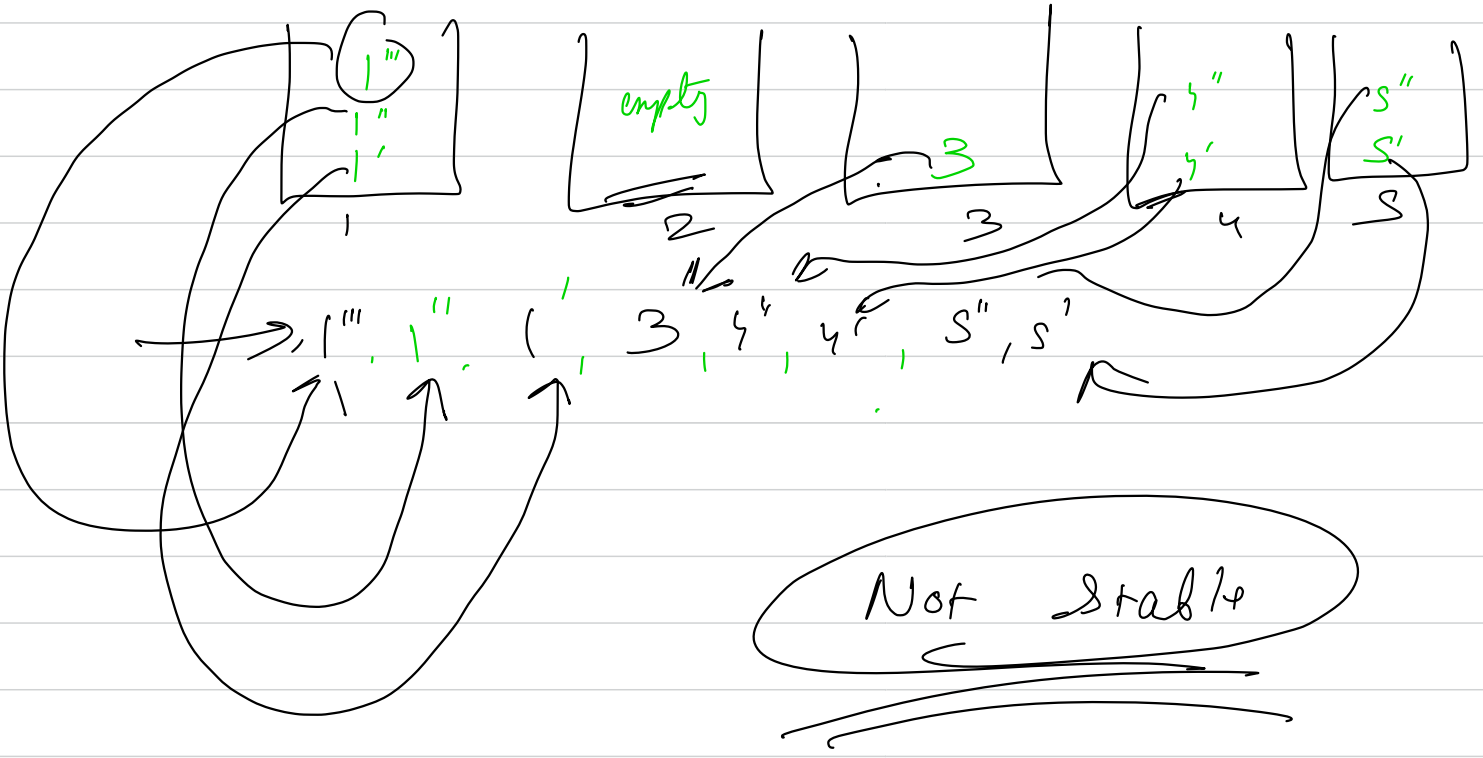
Stable → Yes




5 4 3 2 1
n
n-1
n-2
n-3
1 + n-1 + n-2 - - - - - - 1
O(n²)
5 4 3 2 1 1

Counting Sort

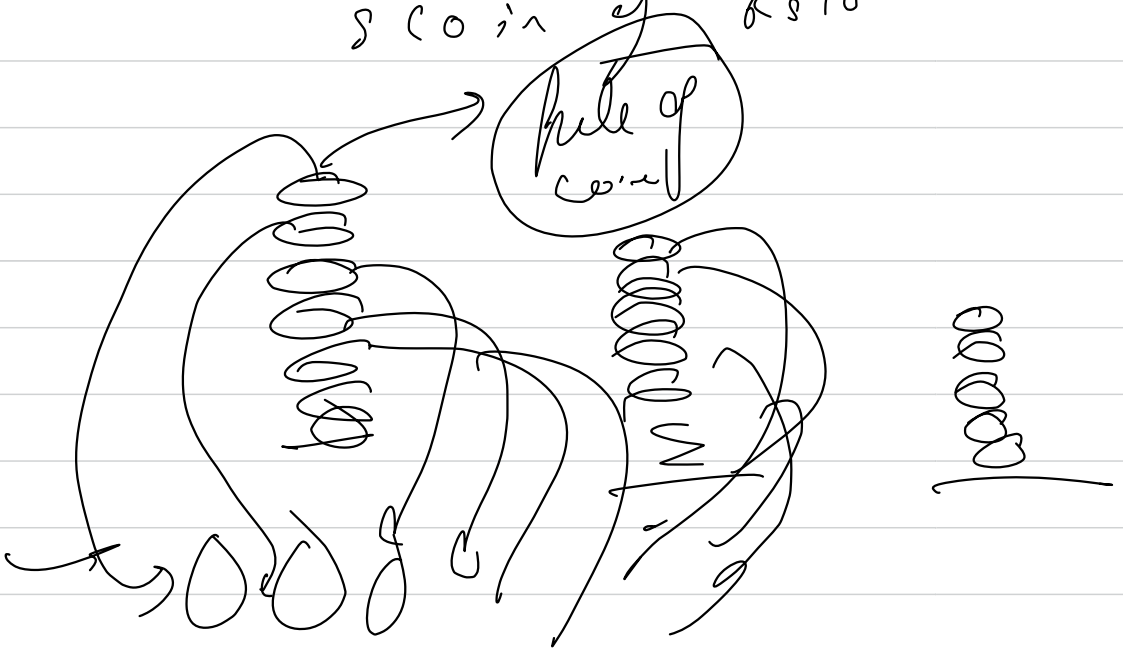



1'''
1''
1'
empty
3
4''
4'
S''
S'
1
2
3
4
S
1''' 1'' 1' 3 4'' 4' S'' S'
Not Stable

Coins → 10 coin of Rs 1

10 coin of Rs 2

5 coin of Rs 10

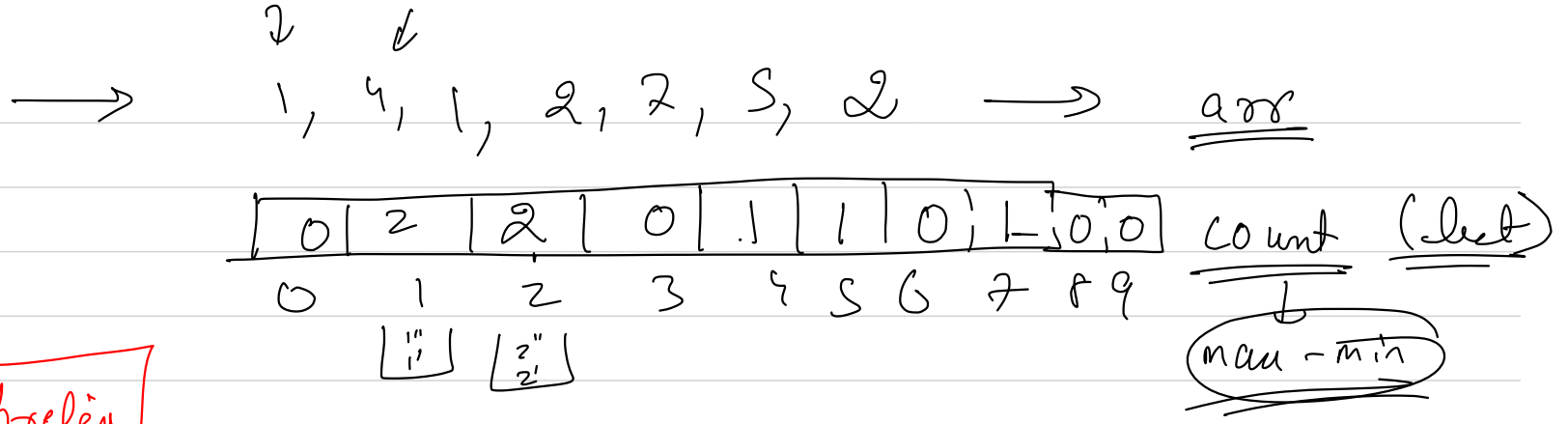


for ——— n

for ——— max − min ——→ k

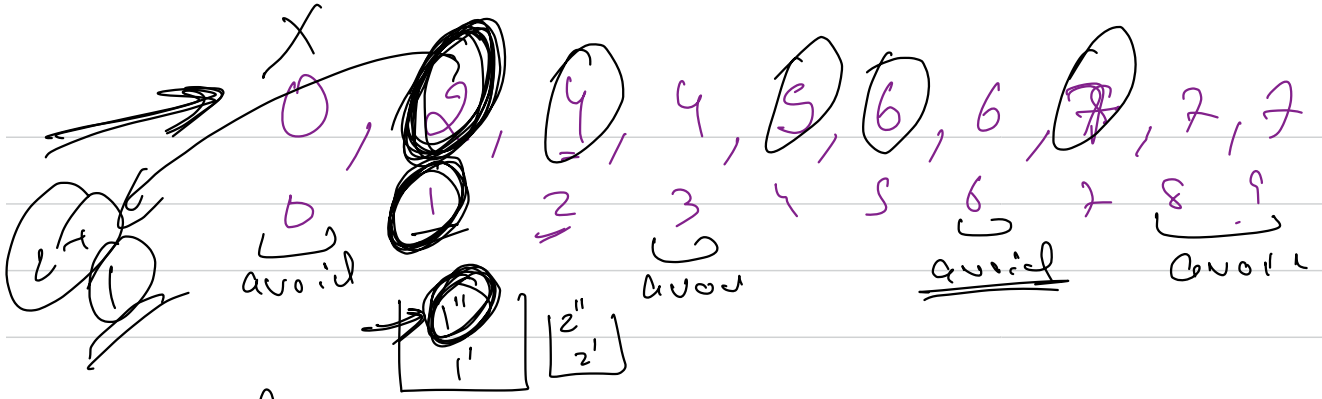
$O(n + (max - min))$

$O(n + k)$



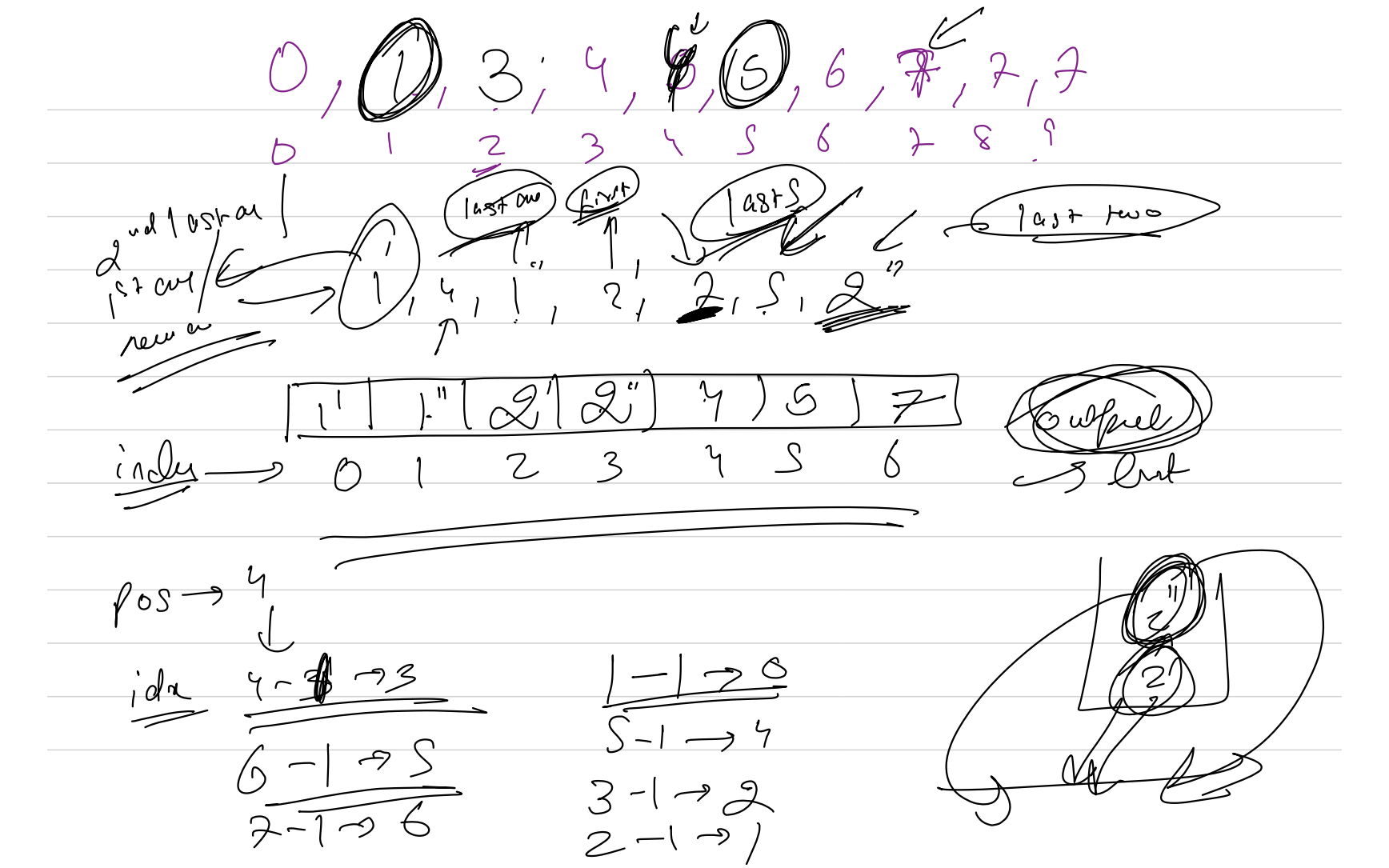
Consider 2 → current value is 4 which denotes the position of the last 2 in the sorted list

→ value of prefix[i] → the position of the last arr[i] in the sorted form.



→ final sorted list →

Y		1	1	2	2	4	5	7
	posin	1	2	3	4	5	6	7


0, 1, 3, 4, 4, 5, 6, 7, 2, 7
0 1 2 3 4 5 6 7 8 9
2nd last
1st one
new one
last one
first
last
last two
1' 1" 2' 2" 4 5 7
index
0 1 2 3 4 5 6
output
pos → 4
idx
4-1 → 3
6-1 → 5
2-1 → 6
1-1 → 0
5-1 → 4
3-1 → 2
2-1 → 1
1"
2
2'

Stable → Yes

Inplace → NO

Space complexity $O(n + K)$

K → range of elements value

TC → $O(n+k)$

Swaps ? → NO

Comparison → NO