Sorting ↳ Ordering

Ascending

(Non-decreasing)

✓ strictly increasing

Descending

(Non-increasing)

A → 1 2 3 4 8 12

B → 1 2 2 3 4 8 12

Strictly decreasing 7 6 4 3 3 2

arr[i] >= → 7 6 4 6

1 4 _ _

Sort

Ascending

Descending

$\boxed{8 > 2}$

[2, 8, 3, 4, 7]

[illegible]

data

'A' < 'Z'

65

[2, 3, 4, 7, 8]

Sorting

→ integers

→ characters

→ Strings

[@, &, }, {]

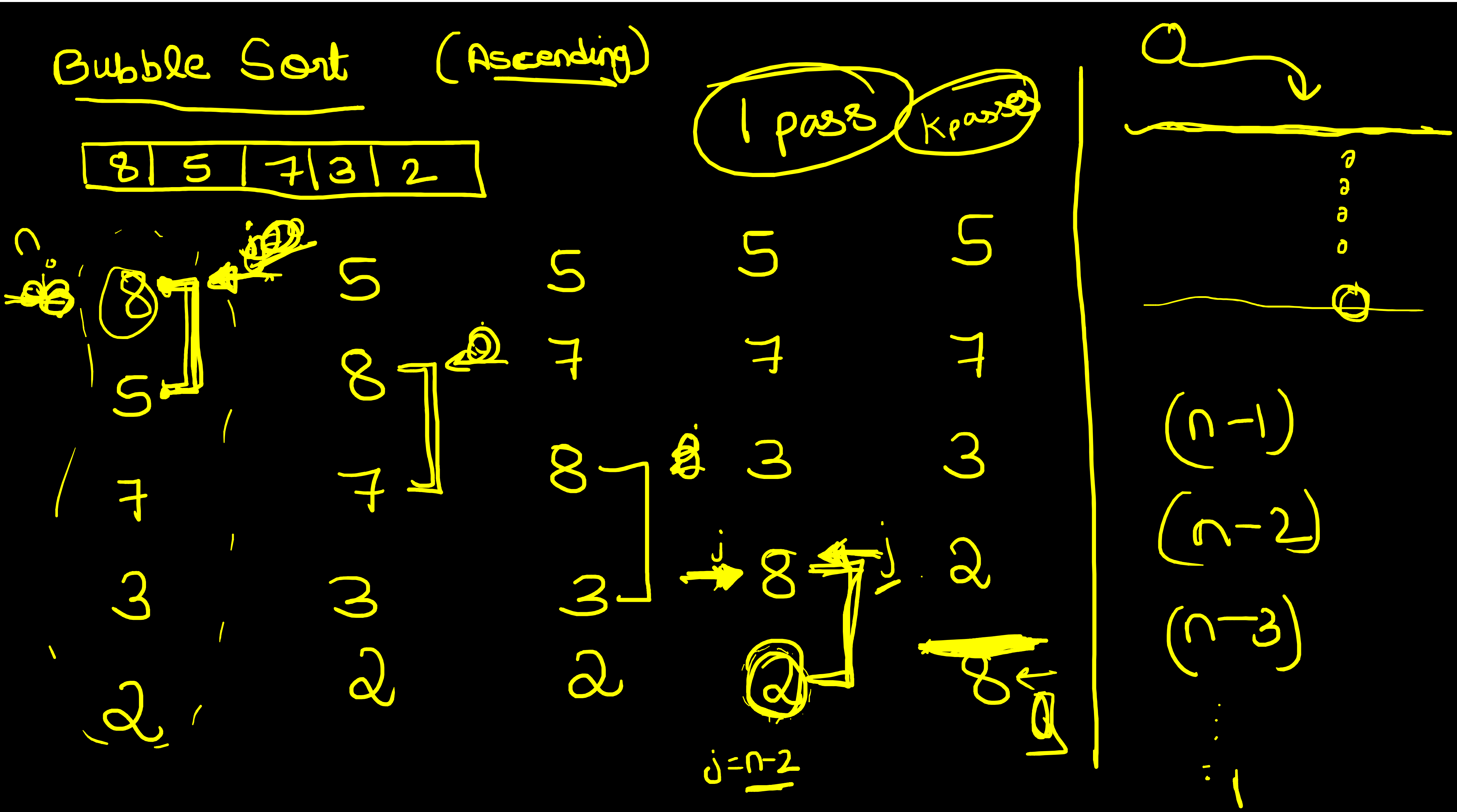
["And", [illegible]]

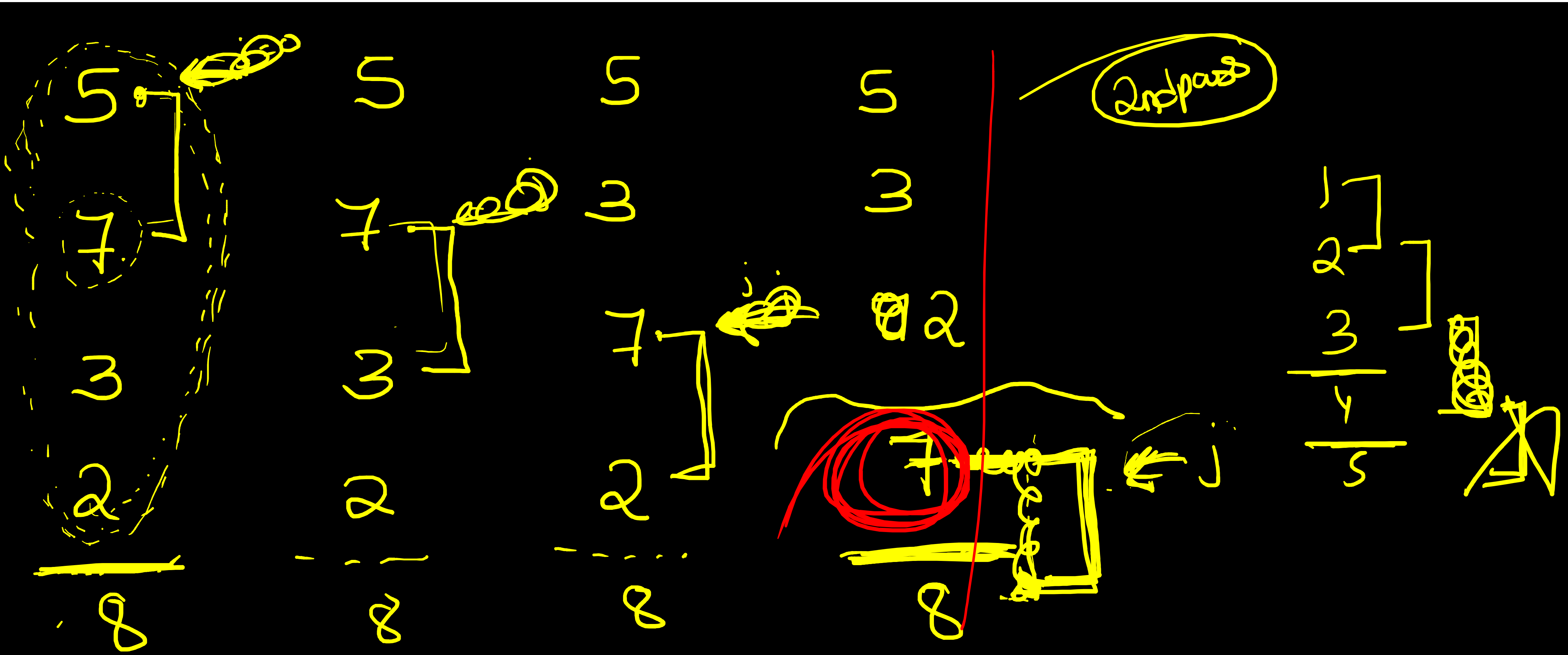
Sorting?

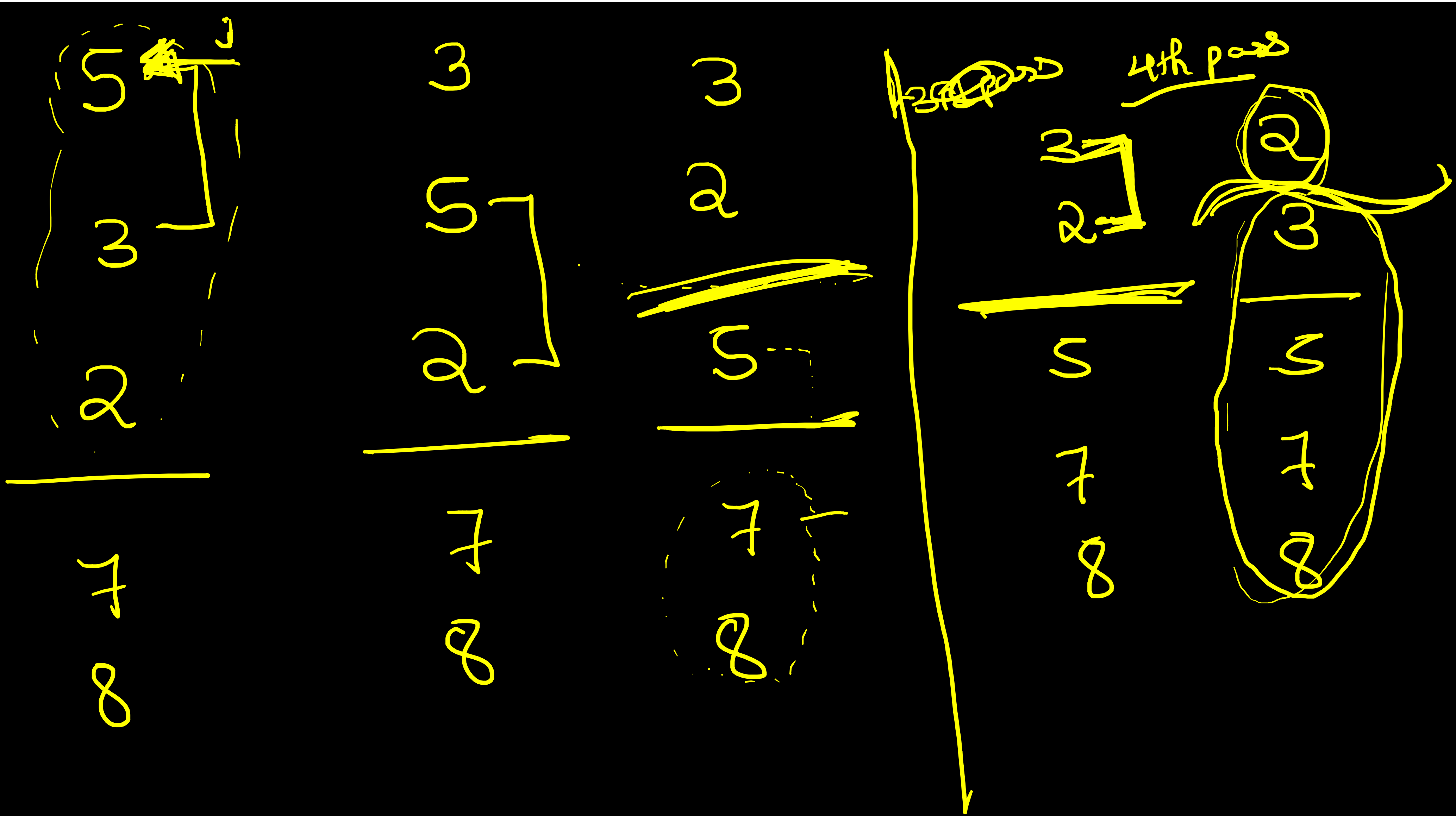
T.C

T.C ↓

Bubble Sort (Ascending)

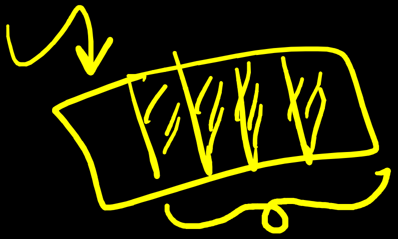



5
7
3
2
8
5
7
3
2
8
5
3
7
2
8
5
3
2
7
8
2ndpass
j
j
1
2
3
4
5


j
5
3
2
7
8
3
5
2
7
8
3
2
5
7
8
3rd pass
3
2
5
7
8
4th pass
2
3
5
7
8

✓ ✓ ✓ ✓ ✓

$n \longrightarrow$ $(n-1)$ passes of Bubble sort



4 passes

$$1+2+3 \;\cdots\; (n-1)$$

$$= \frac{n(n-1)}{2}$$

↳ arr, n

0, 1, 2, 3, ... n-2

```
for (int i=0; i<n-1 ; i++)  → n-1 times
{   → ith pass of bubble sort
    for(int j=0 ; j<n-1-i ; j++)
    {
        if ( arr[j] > arr[j+1])
        { swap (arr[j], arr[j+1]);}
    }
}
```

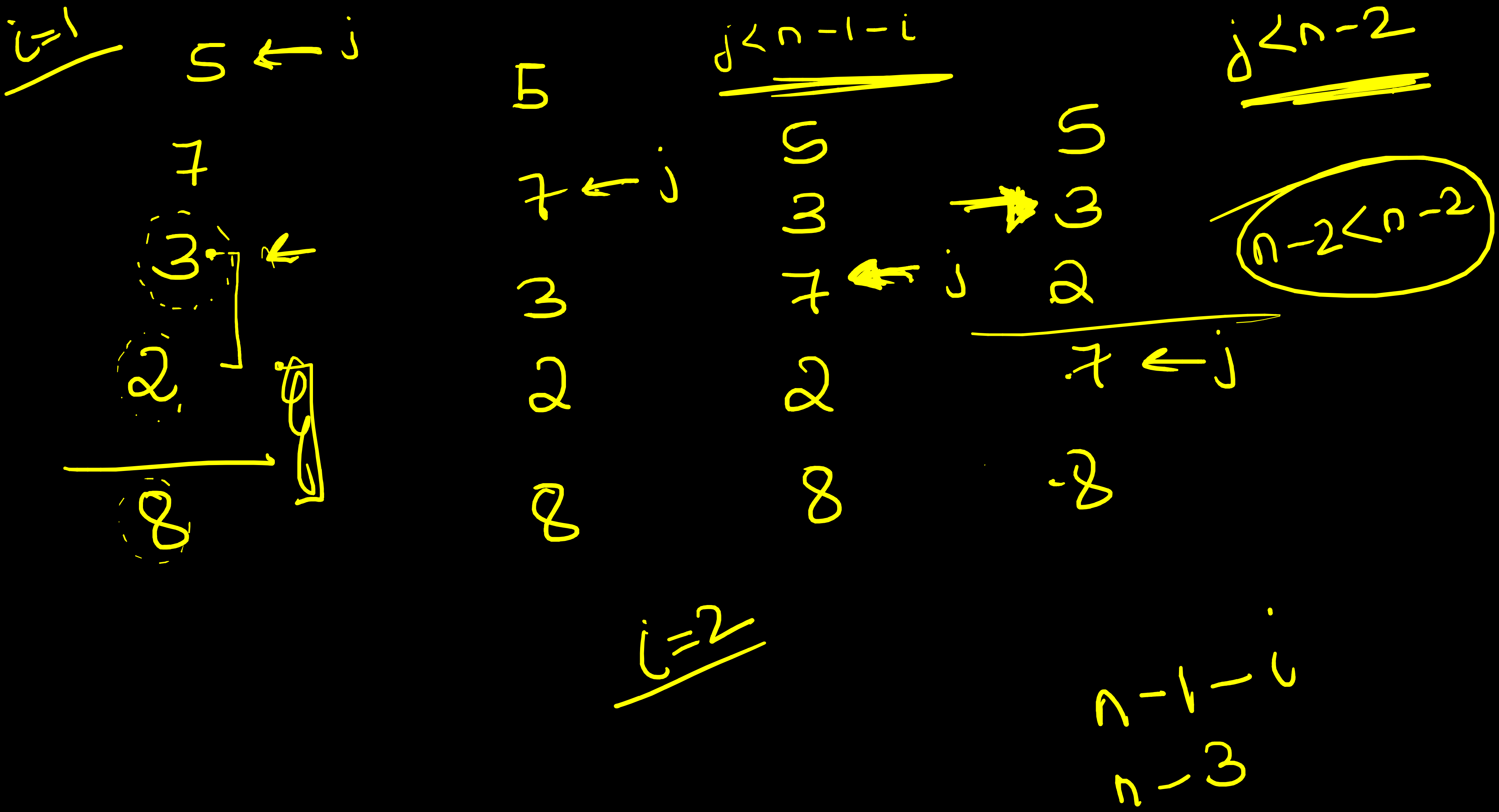
$\underset{0}{} \; \overbrace{1, 2, 3, 4, \; n-2}^{n-2}$

n-2

n-1-1

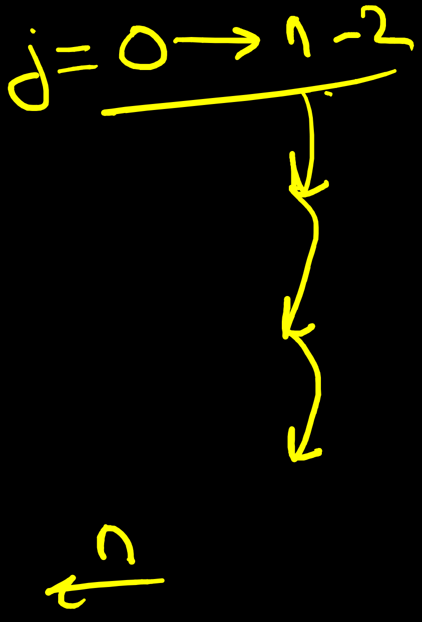
i=0

```
 8 ←j     5        5       5       5
 5        8 ←j     7       7       7
 7        7        8 ←     3       3
→3        3        3       8 ←j    2
 2        2        2       2       8 ←j
```

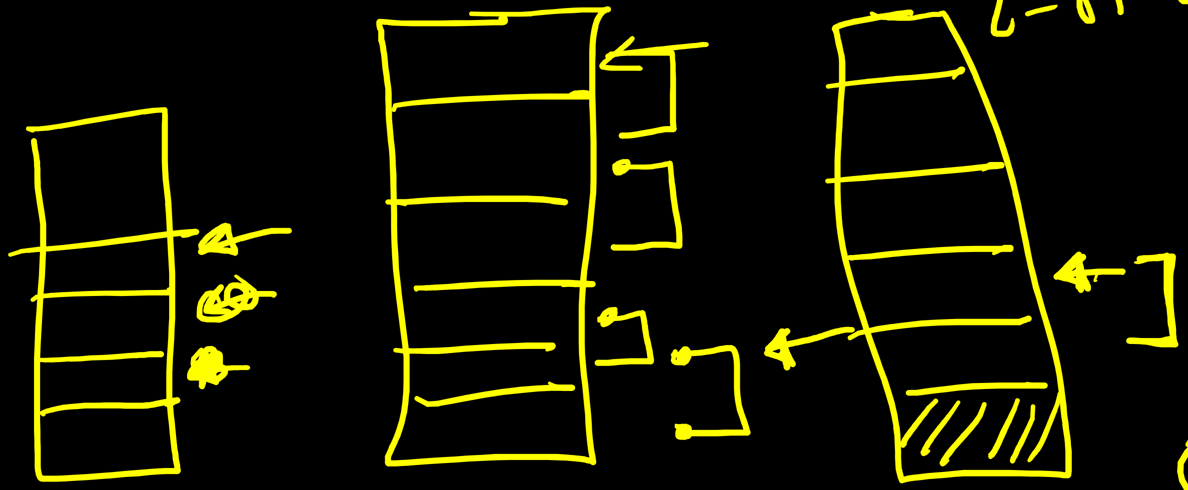

i=1
5 ← j
7
3
2
8
5
7 ← j
3
2
8
j<n-1-i
5
3
7 ← j
2
8
j<n-2
5
3
2
7 ← j
8
n-2<n-2
i=2
n-1-i
n-3

```
for ( int i= 1; i<= n-1 ; i++)
{     // ith pass of bubble sort
      for(int j=0; j<n-i ; j++)
      {

      }
}
```



```
for (i=0 ;i<n-1 ; i++)
{ for( int j=0 ; j<n-1-i; j++ )
    { }
}
```



$i=0 \longrightarrow n-1$ times

$i=1 \longrightarrow n-2$ times

$i=2 \longrightarrow n-3$

⋮

$i=n-2 \longrightarrow 1$ times

$$\frac{n(n-1)}{2}$$

$$O(n^2)$$

$\boxed{O(n^2)}$

$O(1)$

$\frac{n(n-1)}{2}$

① Find 2 long

② Find k th largest arr[n-k]

Stable

$[1, 2, 8^{A}, 7, 8^{B}]$

A, B

$[B, A]$

$\rightarrow [1, 2, 7, 8^{B}, 8^{A}]$

unstable

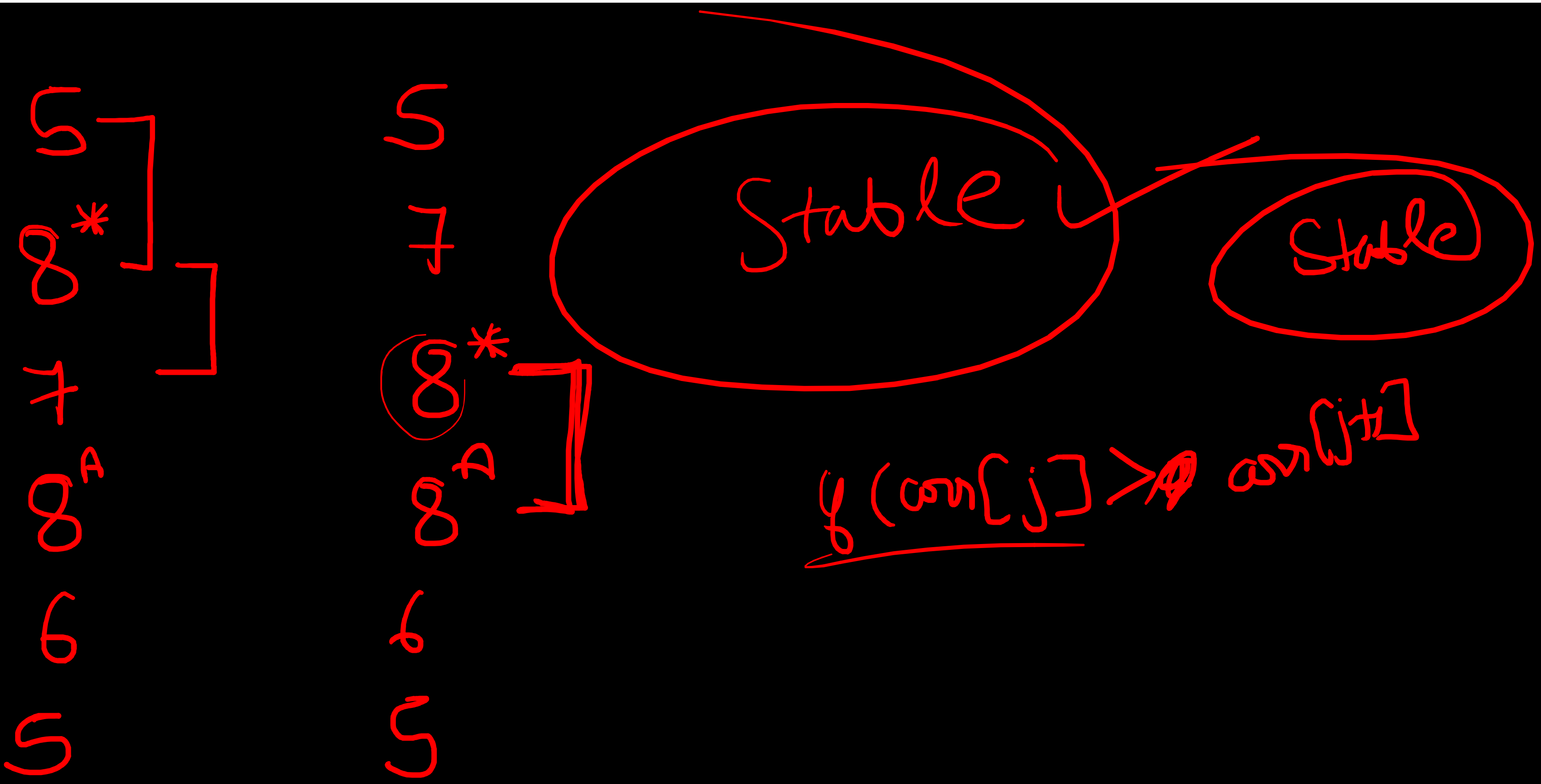
$\rightarrow [1, 2^{A}, 2^{D}, 3^{\alpha}, 2^{E}, 3^{B}]$

Struct

$\Rightarrow [1, 2^{A}, 2^{D}, 2^{E}, 3^{\alpha}, 3^{B}]$

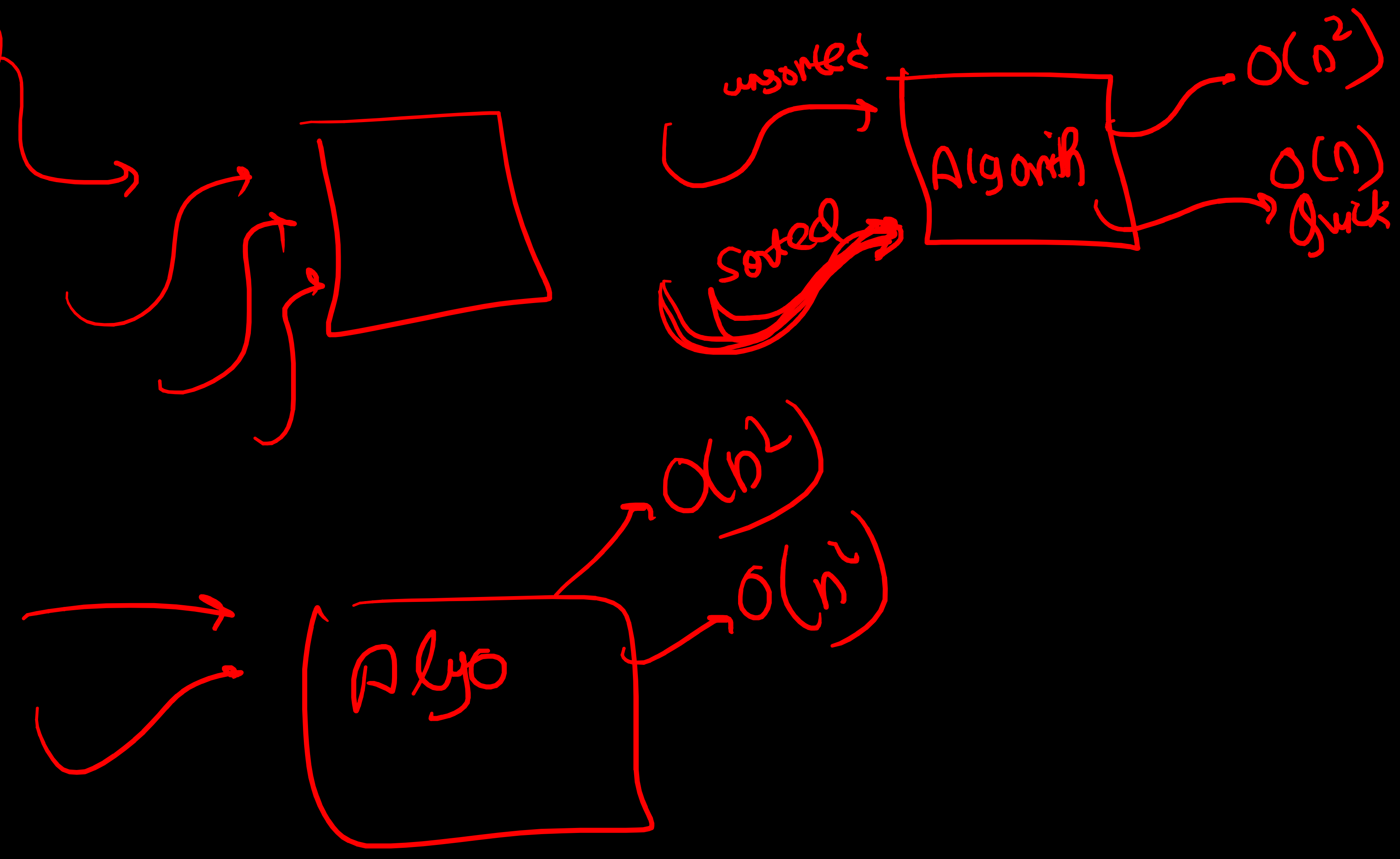
$[1, \boxed{2}^{A}, \boxed{2}^{D}, 3^{\alpha}, \boxed{2}^{E}, 3^{B}]$

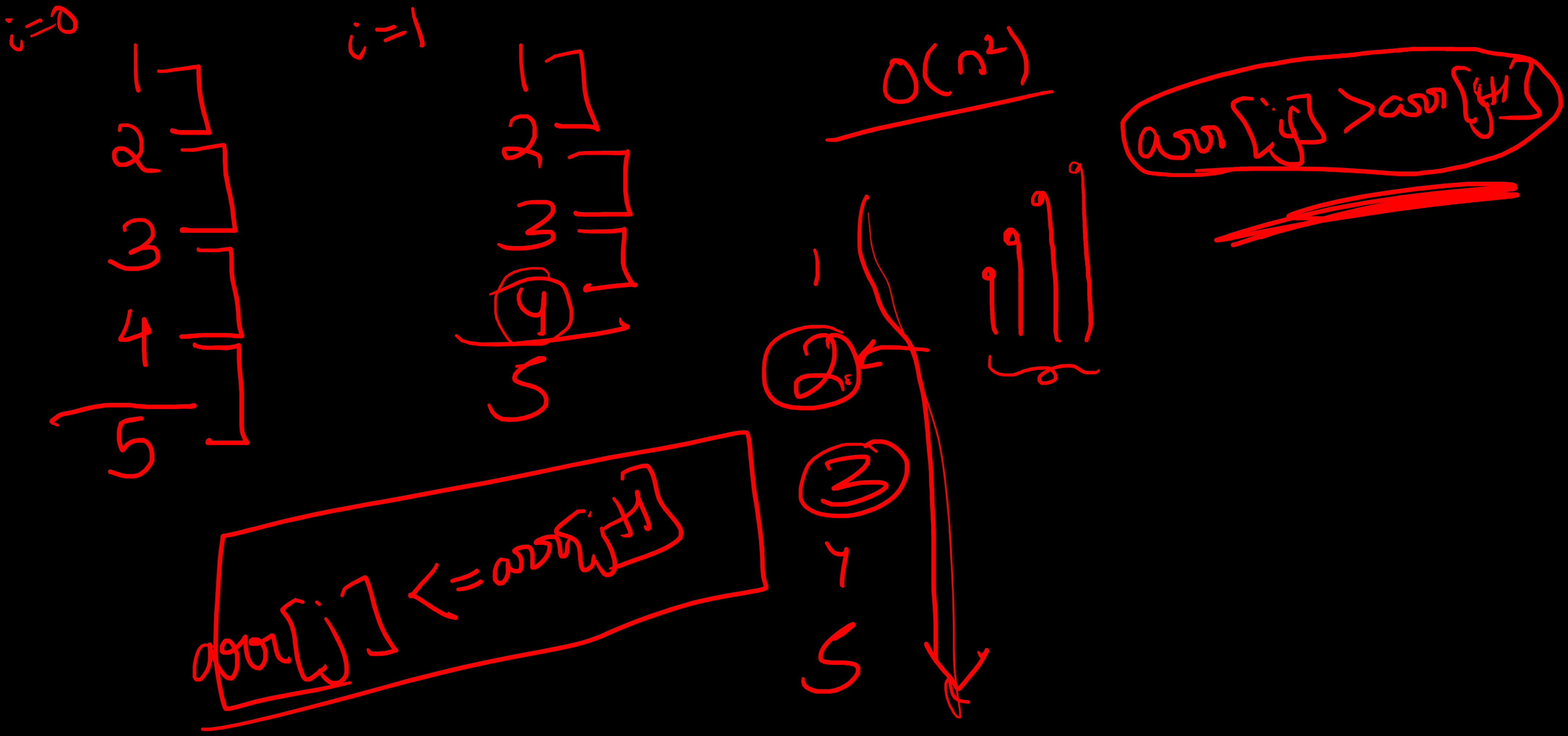
$[$



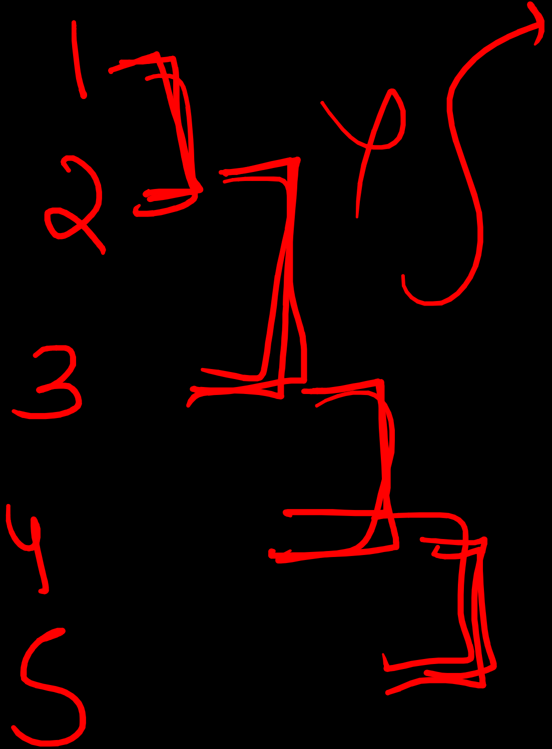
if (arr[j] > arr[j+1])

Adaptive




i=0
1
2
3
4
5
i=1
1
2
3
4
5
O(n^2)
arr[j] > arr[j+1]
1
2
3
4
5
arr[j] <= arr[j+1]

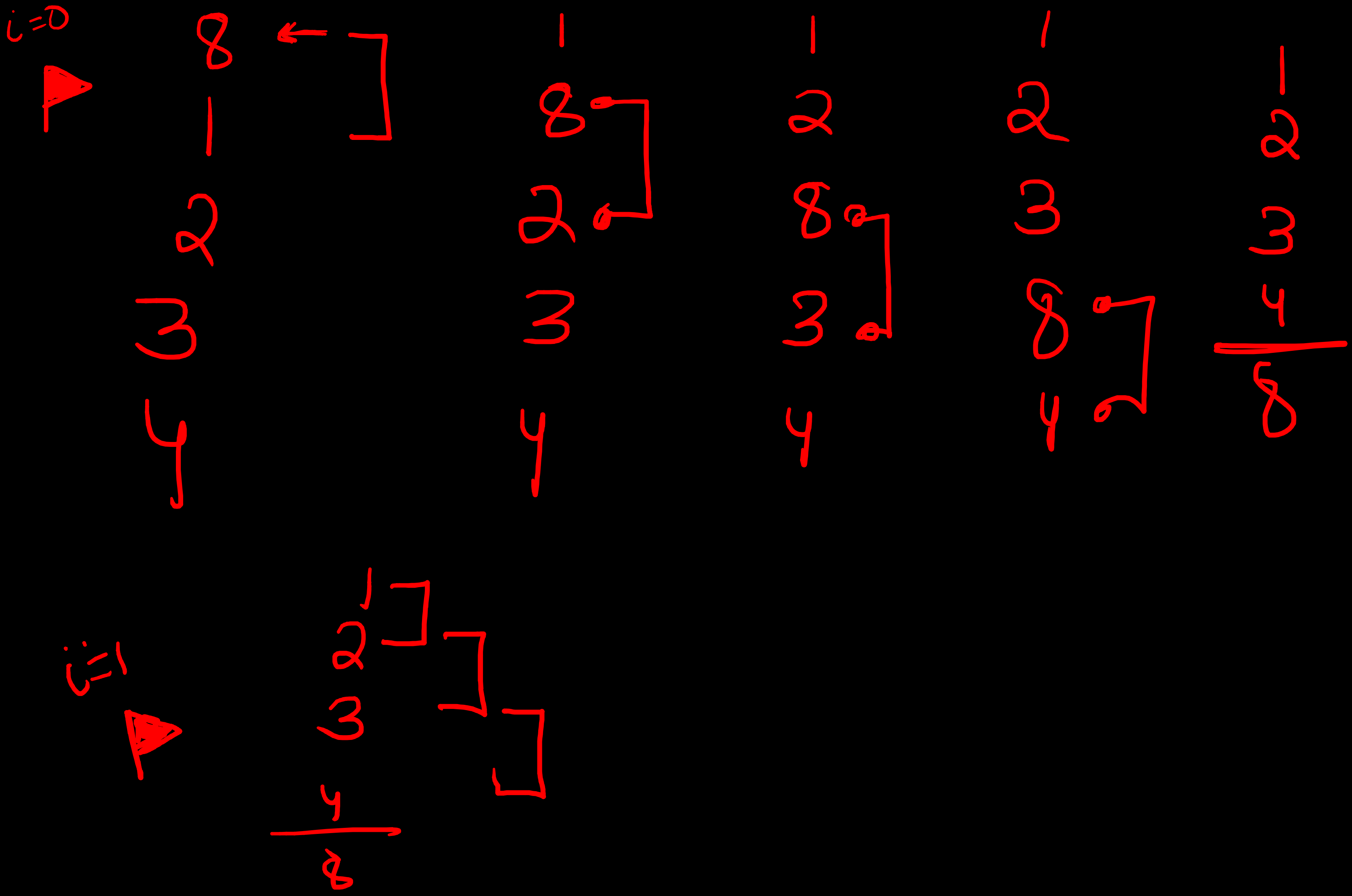
i=0



sorted

O(n)

(n-1) comparisons


i=0
8
1
2
3
4
1
8
2
3
4
1
2
8
3
4
1
2
3
8
4
1
2
3
4
8
i=1
1
2
3
4
8