| 70 | 50 | 60 | 10 | 20 | 30 | 80 | 15 |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

Property of inversion

$$\boxed{\begin{array}{l} i < j \text{ — (1)} \\ arr(i) > arr(j) \text{ — (2)} \end{array}}$$

→ Definition of Inversions

70 → 50, 60, 10, 20, 30, 15

50 → 10, 20, 30, 15

60 → 10, 20, 30, 15

10 →

20 → 15

30 → 15

80 → 15

15 →

\# inversions = 17

# Brute force Approach

i < j
arr(i) > arr(j)

```
inversion( arr, n) {
    for (i = 0 to n-1) {
        for( j = i+1 to n) {
            if(arr(i) > arr(j))
            {
                count++;
            }
        }
    }
    return count
}
```

i < j

arr(i) > arr(j)

Time complexity → $O(n^2)$

Decreasing order

| 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 5 | 4 | 3 | 2 | 1 |

5 —— 4, 3, 2, 1

4 —— 3, 2, 1

3 —— 2, 1

2 —— 1

1 ——

Maxima

# inversions = 10

Increasing order

| 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |

$i < j$

$arr(i) > arr(j)$

1 ____

2 ____

3 ____

4 ____

5 ____

Minima

$\boxed{\#\ \text{inversions} = 0}$

Pre-requisite

Mergesort

↳ MergeProcedure

Optimized Approach

Divide & conquer

| 0 |
|---|
| 15 |

→ 0

Small Problem

Single element

↳ # inversions = 0

Big Problem

↳ 1) Divide

c ____ $mid = l + (h-l)/2$

2) conquer

$2T(n/2)$ Left subarray ↳ inversion(arr, l, mid)

Right subarray inversion(arr, mid+1, h)

3) combine

→ Merge procedure

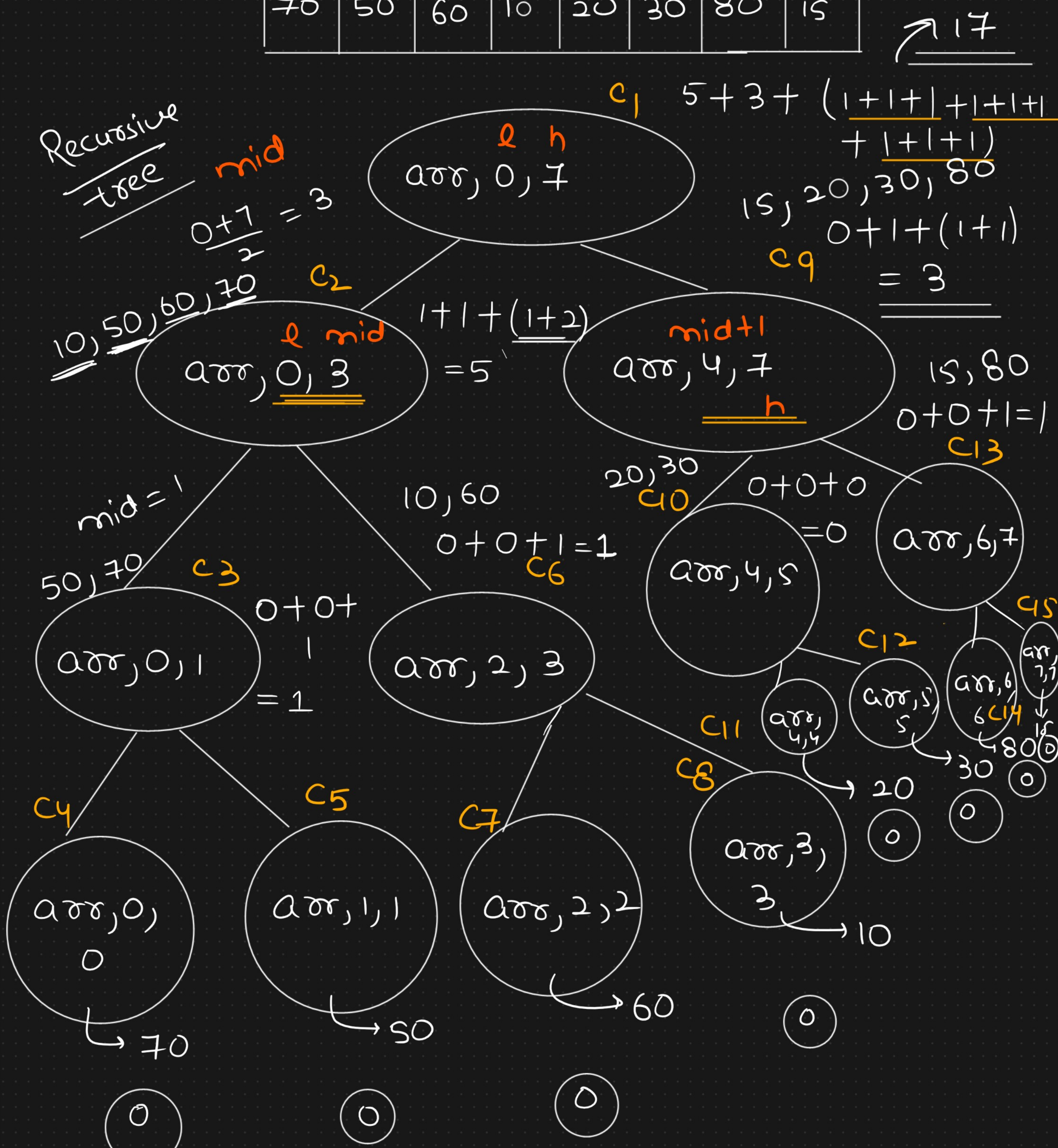
(count the swaps)

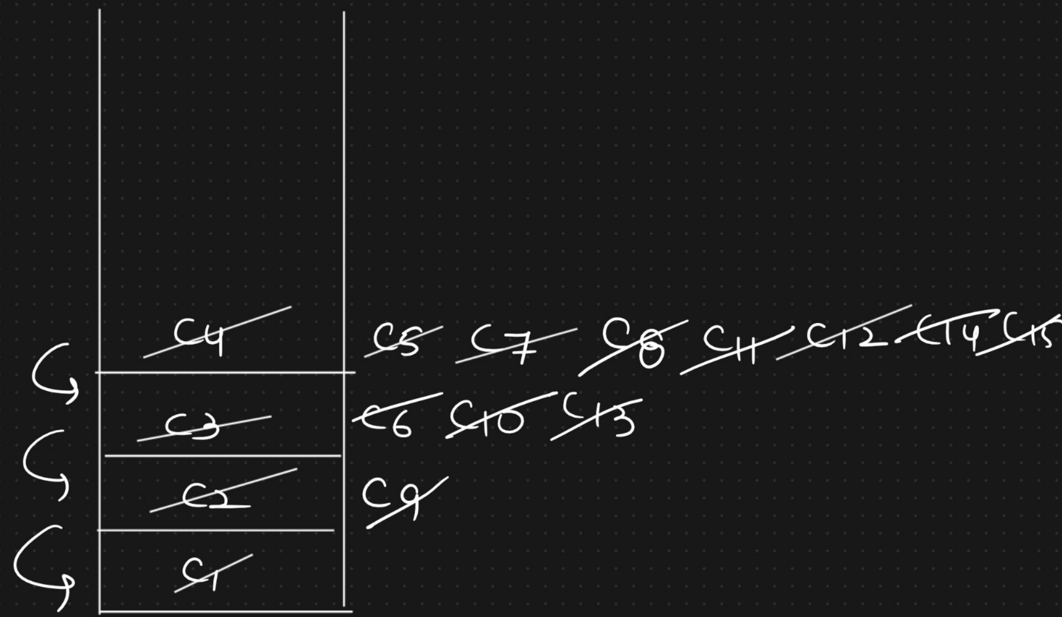
Assignment

swap ??

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| 70 | 50 | 60 | 10 | 20 | 30 | 80 | 15 |

Recursive tree






## Recurrence Relation

$$T(n) = \begin{cases} 1 & n = 1 \\ 2T\left(\frac{n}{2}\right) + n & n > 1 \end{cases}$$

$$\boxed{T(n) = O(n \log n)}$$

$$\boxed{\text{Space complexity} = O(n)}$$

Extra space
(MergeProcedure)