Let's start at 9:02 PM

L30
Merge Sort and more

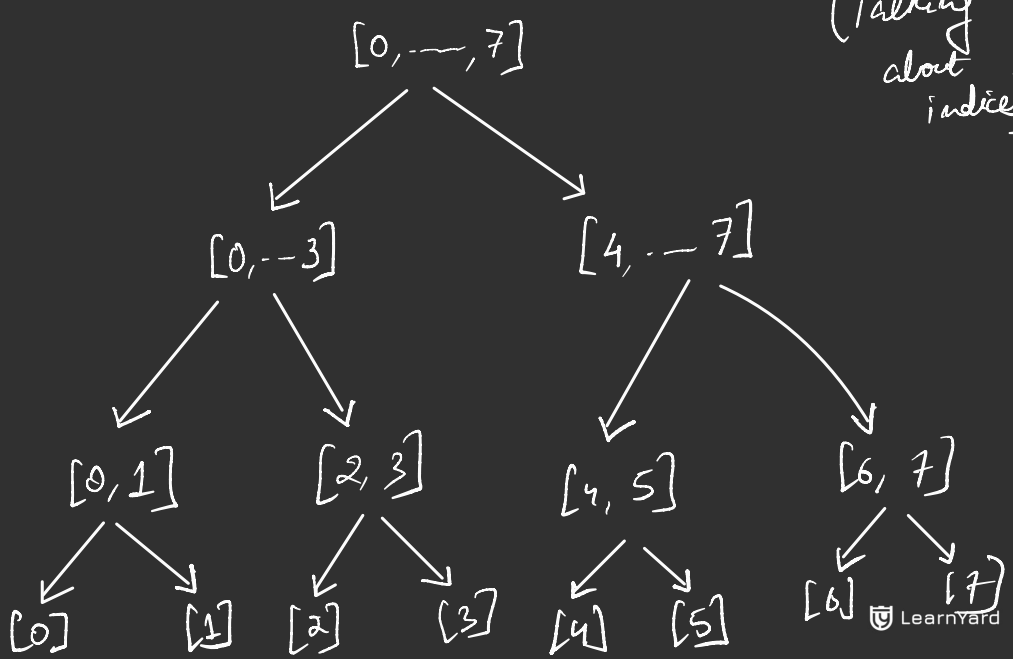
If interested, check out the System Design course.
Early Bird discount is ON.

Join Discord - https://bit.ly/ly-discord

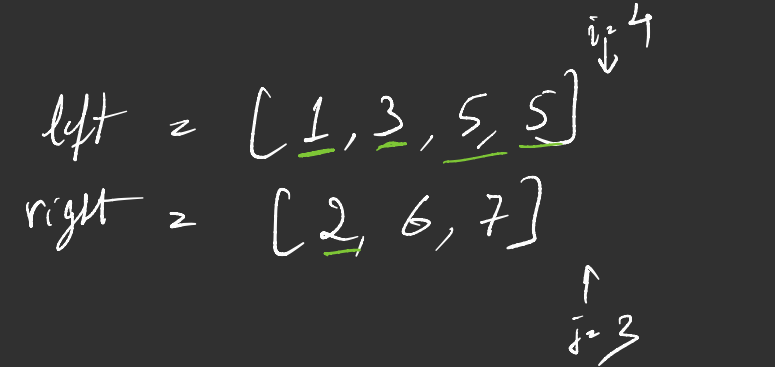
RECAP

Remember Insertion Sort?

Let's come to Merge Sort



Merge 2 sorted Arrays

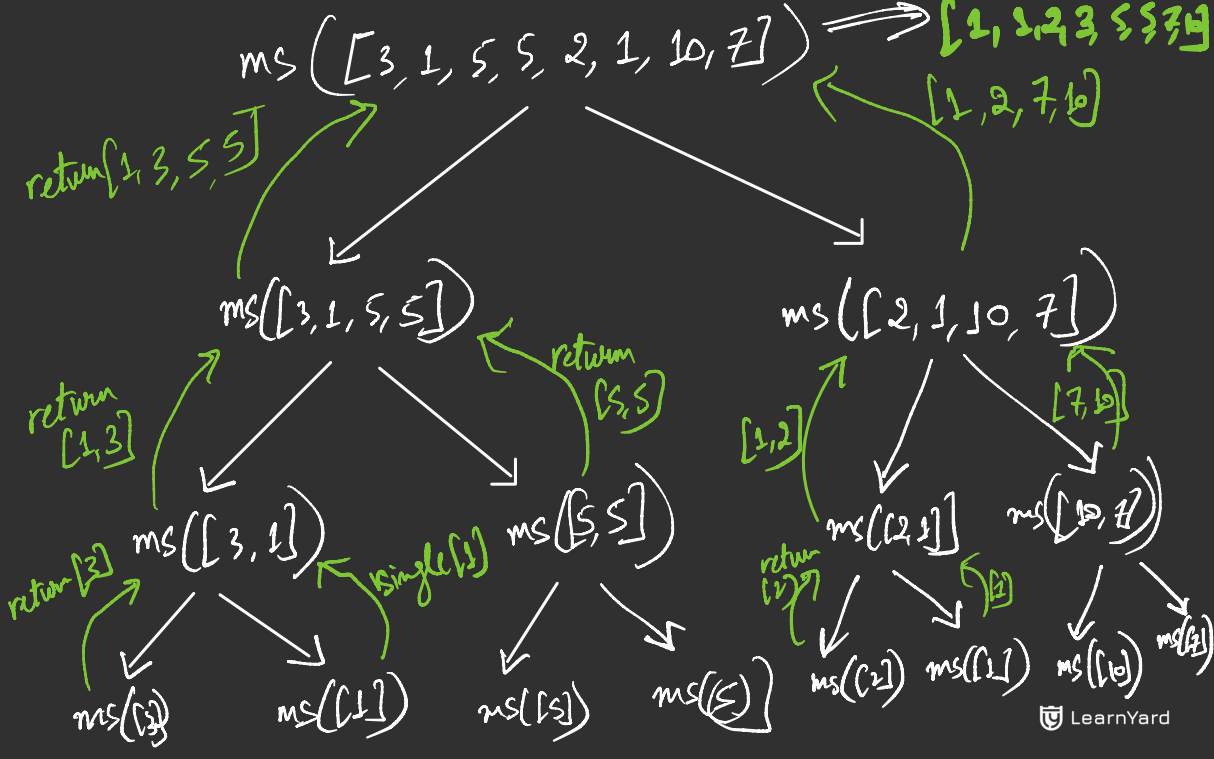


ans = [1, 2, 3, 5, 5, 6, 7]

pseudo-code

mergeSorted(—)

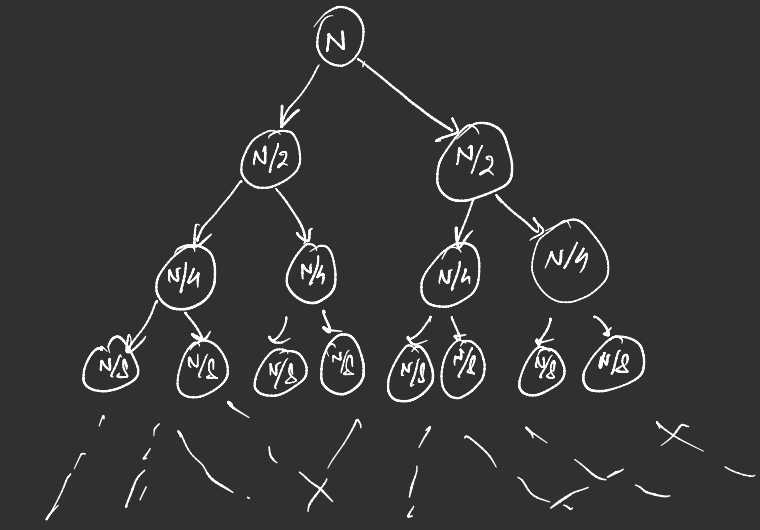
```
int[] ms(arr, n) {
    if (n == 1)
        return arr;

    int[] left ⇒ equal to the first half of arr.
    int[] right ⇒ equal to the second half of arr.

    left = ms(left, size(left));
    right = ms(right, size(right));

    return merge2sorted(left, right);
}
```


ms([3,1,5,5,2,1,10,7])
[1,1,2,3,5,5,7,10]
return [1,3,5,5]
[1,2,7,10]
ms([3,1,5,5])
ms([2,1,10,7])
return [1,3]
return [5,5]
[1,2]
[7,10]
return [3]
ms([3,1])
single [1]
ms([5,5])
ms([2,1])
ms([10,7])
return [1]
[1]
ms([3])
ms([1])
ms([5])
ms([])
ms([2])
ms([1])
ms([10])
ms([7])
LearnYard

Let’s Implement

Trying using recursion tree

Let's understand the time complexity



$$T(N) = 2 * T(N/2) + N$$

$$2 * T(N/2) = 4 * T(N/4) + N$$

$$4 * T(N/4) = 8 * T(N/8) + N$$

$$\vdots$$

$$K * T(2) = 2K * T(1) + N$$

$\log_2 N$

$$T(N) = \underbrace{N + N + N + N - - - - N}_{\log_2 N}$$

$$T(N) = O(N * \log N)$$

1. What is the total number of levels in the recursion tree while using merge sort on an array of 8 elements?

A. 1
B. 2
C. 3
D. 4

1. What is the total number of levels in the recursion tree while using merge sort on an array of 8 elements?

A. 1
B. 2
C. 3
D. 4

Solution :

[X X X X X X X X] - Level 0

[X X X X] - [X X X X] - Level 1

[X X] - [X X] - [X X] - [X X] - Level 2

[X]-[X]-[X]-[X]-[X]-[X]-[X]-[X] - Level 3

2. What is the total number of nodes in the recursion tree while using merge sort on an array of 4 elements?

A. 3
B. 7
C. 9
D. 12

2. What is the total number of nodes in the recursion tree while using merge sort on an array of 4 elements?

A. 3
B. 7
C. 9
D. 12

Solution :
[X X X X] -> level 0 : nodes 1
[X X] - [X X] -> level 1 : nodes 2
[X]-[X]-[X]-[X] -> level 2 : nodes 4
Total Nodes = 1 + 2 + 4 = 7

3. What is the auxiliary space complexity of the merge sort algorithm discussed?

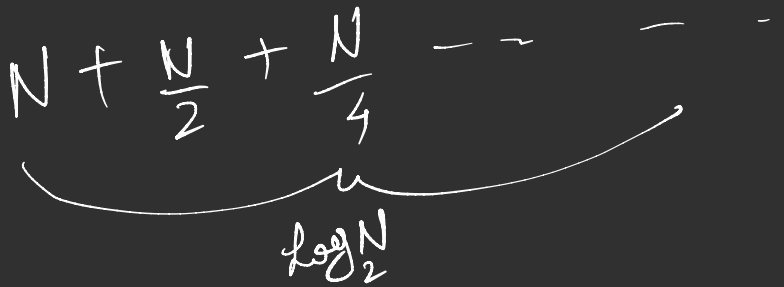
A. O(1)
B. O(n)
C. O(n log n)
D. O(log n)

3. What is the auxiliary space complexity of the merge sort algorithm discussed?

A. O(1)
B. O(n)
C. O(n log n)
D. O(log n)

Solution : We need an array of length **n** to merge two arrays.

Let's understand space complexity



space ⇒ O(N)

4. Can we parallelly sort two partitions in merge sort to speed up the process? 💡

A. False
B. True

4. Can we parallelly sort two partitions in merge sort to speed up the process? 💡

A. False
B. True

Solution : Since two portions our independent of each other we can start sorting both the portions at the same instant to save time.

Let’s solve 1 problem? xD

Given an array, find its inversion count.

[2, 4, 1, 3, 5]

(2, 1)
(4, 1)
(4, 3)

3

Implementation is homework.
(Codes will be shared in both C++ and Java)

Thank You!

Reminder: Going to the gym & observing the trainer work out can help you know the right technique, but you'll muscle up only if you lift some weights yourself.

So, PRACTICE, PRACTICE, PRACTICE!