Time Complexity for Recursion

Monday, 19 February 2024 8:45 PM

```
for(int i=0; i<n; i++){
    // Operation
    // operation
}
```

$\Rightarrow 2n$

$f(n) = \Sigma$ (operation) ??

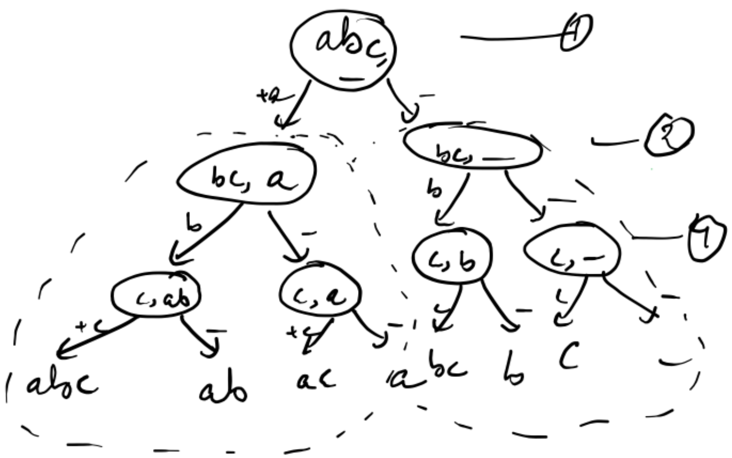
??

Recursion Tree's height $\Rightarrow$ No. of operations ??

↳ At a particular level did we perform

Subsequence

abc



$T(n) = T(n-1) + T(n-1) + n$

$T(n) = 2T(n-1) + n$ $\boxed{2^n \times n}$ $\Rightarrow T(n) = 2T(n-1) + n$

$$T(n-1) = 2T(n-2) + (n-1) \quad \times 2$$

$$2T(n-1) = 4T(n-2) + 2(n-1)$$

$$T(n-2) = 2T(n-3) + (n-2)$$

$$T(n) = 4T(n-2) + n + 2(n-1)$$

$$T(n-2) = 2T(n-3) + (n-2) \quad \times 4$$

$$4T(n-2) = 8T(n-4) + 4(n-2)$$

$$\vdots$$

$$T(1) = 1$$

Factorial

$$n! = n \cdot n-1 \cdot n-2$$

$$\text{factorial}(n) = n \times \text{factorial}(n-1)$$

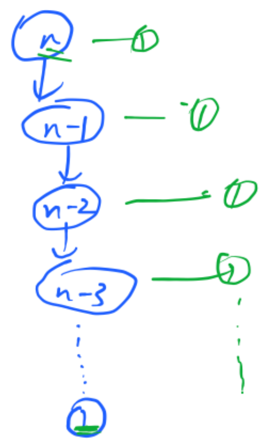
$$T(n) = T(n-1) + 1$$

$$T(n-1) = T(n-2) + 1 \qquad f(n-1) = f(n-2) \cdot (n-1)$$

$$T(n-2) = T(n-3) + 1$$

$$\vdots$$

$$T(1) = 1$$



$$(1) \times n$$

$$T(n) + \cancel{T(n-1)} + \cancel{T(n-2)} + \cdots + \cancel{T(1)} = \cancel{T(n-1)} + 1 + \cancel{T(n-2)} + 1 + \cancel{T(n-3)} + 1 + \cdots + 1$$

$$T(n) = 1 + 1 + 1 + \ldots \text{ n times}$$

$$\rightarrow \underline{\underline{n}} \rightarrow O(n)$$

3^n

4^n

Fibonacci

$$fibo(n) = fibo(n-1) + fibo(n-2)$$

$$T(n) = \underbrace{T(n-1) + T(n-2)} + 1$$

$$< 2T(n-1) + 1$$

$$T(n) = \underline{2}T(n-1) + 1 \longrightarrow 2^n$$

$\underline{\underline{n}}$

$$T(n) = 2T(n-1) + 2^0$$

$$2T(n-1) = 4T(n-2) + 2^1$$

$$4T(n-2) = \underbrace{8\ T(n-3)} + 4 \rightarrow 2^2$$

$$T(n) = 2^0 + 2^1 + 2^2 + \ldots 2^{n-1}$$

$$= \frac{2^n - 1}{2 - 1}$$

$$= \underline{2^n - 1}$$

$$\sim O(2^n)$$

$T(1) = 1 + 1 \times 2^{n-1}$ ~ $O(\underline{n})$

Monday, 19 February 2024
9:09 PM

Merge Sort

$\underline{A1} = \{10, 30, 50, 80, 100\}$ 5 → 2 sorted arrays
$\underline{A2} = \{15, 20, 25, 40, 90\}$ 5 →

⇓

$\underline{\underline{A3}} = \{10, 15, 20, 25, 30, 40, 50, 80, 90, 100\}$

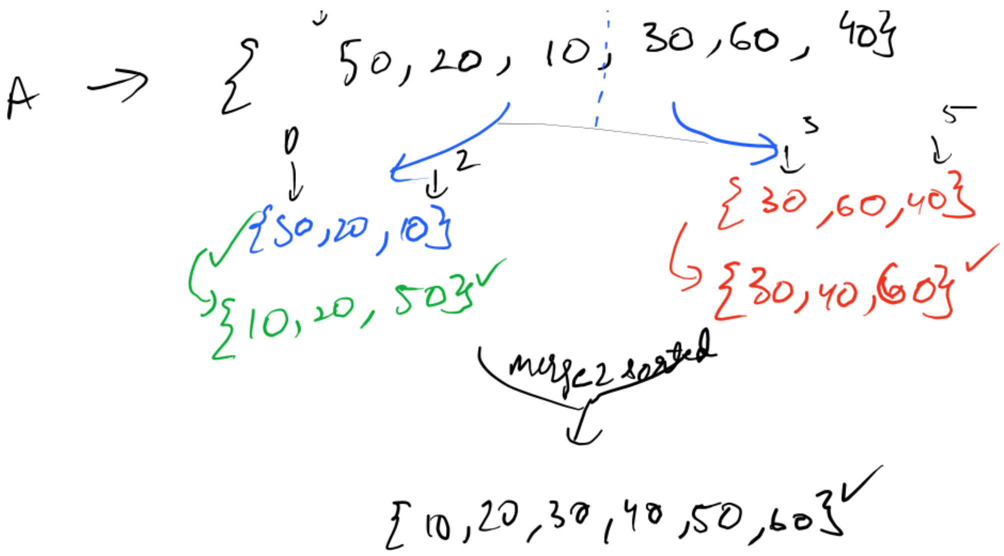
sort

sorted✓

(1) [10 30 50 80 100 15 20 25 40 90]

$\underline{A1} = \{\cancel{10}, \cancel{30}, \cancel{50}, \cancel{80}, 100\}$
$A2 = \{\cancel{15}, \cancel{20}, \cancel{25}, \cancel{40}, 90\}$

$\{10, 15, 20, 25, 30, 40, 50, 80, 90, 100\}$

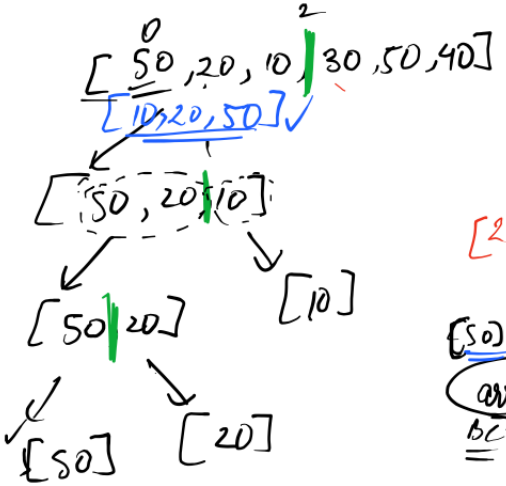


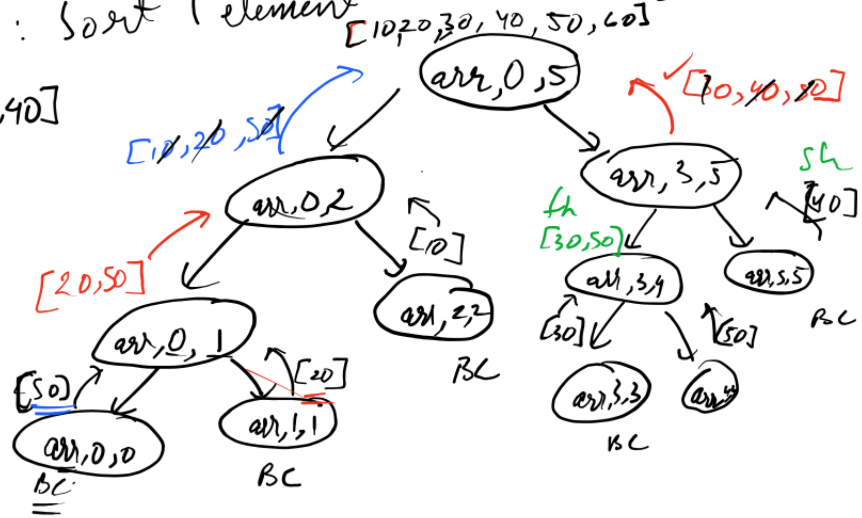
BP: Sort n elements

SP: Sort n/2 elements

- SW: → merge 2 sorted arrays

BC: Sort 1 element





$n/2 \quad n/2$

$\rightarrow n$

$\boxed{[100]}$ ✓

th sh

$T(n) = T(n/2) + T(n/2) + n$

↳ merge 2 sorted

$T(n) = 2T(n/2) + n$

↳ $O(n \log n)$

Monday, 19 February 2024
10:32 PM

Quick Sort

e s

$\{50, 40, \cancel{60}, \cancel{90}, \cancel{70}, \cancel{10}, \cancel{30}, \cancel{60}, 120\}$

60 30 10 70 90 80

Pivot → 70

↳ steps

1) Find Pivot
2) Partition the array → All the elements smaller than pivot should be on the left, All the greater elements should be on the right.
3) Recursion

↳

e s

$50, 40, \cancel{60}, 30, \cancel{10}, 70, 90, 80, 120$

10 60

pivot=...

pivot=40 {~~50~~, ~~40~~, 10, ~~30~~}

30 10 40 50

logic
- partition (arr, l, r) → SW
- quick (arr, l, e) → SP
- quick (arr, s, r) → SP

BP: Place n ~~elts~~ elements at correct position

SP: Place n-1 elements at correct posn

SW: partitioning

BC: when No elements are under consideration ($r<l$)

{50, 40, ~~80~~, 90, ~~70~~, ~~10~~, 30, ~~60~~, 120}

60 30 10 70 90 80

50, 40, ~~60~~, 30, ~~10~~

10 60

~~50~~, ~~40~~, ~~10~~, ~~30~~

30 10 40 50

60

10, 30, 40, 50, 60

1 2 3 4

0 1 2 3

(0,0)

QS(2,3)

3 4

2 r=3

QS(2,1) $r \le l$

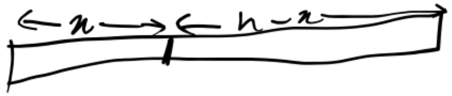
↳ return

Unstable sorting algo

↳ where order isn't retained

Quick → Unstable

Merge → Stable



$$T(n) = T(x) + T(n-x) + n$$

Case 1

$x = 1$

$$T(n) = T(1) + T(n-1) + n$$

$$T(n-1) = T(1) + T(n-2) + (n-1)$$

$$\vdots$$

$$= n \cdot T(1) + n + (n-1) + (n-2) + \dots$$

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

$O(n^2) \rightarrow$ TC

↳ Worst

Case 2

$x = n/2$

$$T(n) = T(n/2) + T(n/2) + n$$

$$= 2T(n/2) + n$$

$$\Downarrow$$

$n \cdot \log n$

↳ Average