Name : S.S. Nadhiya

Reg no : 192321136

Course Code : CSA0669

Date : 06/06/24

Analytical : 01

1. Solve the following recurrence relations :

a) $x(n) = x(n-1) + 5$ for $n > 1$ $x(1) = 0$

$$x(n) = x(n-1) + 5 \quad \text{— } 1$$

$$x(n-1) = x(n-1-1) + 5$$
$$= x(n-2) + 5 \quad \text{— } 2$$

$$x(n-2) = x(n-2-1) + 5$$
$$= x(n-3) + 5 \quad \text{— } 3$$

Sub eq 3 in 2 ;

$$x(n-1) = x(n-3) + 5 + 5$$
$$= x(n-3) + 10 \quad \text{— } 4$$

Sub eq 4 in eq 1

$$x(n) = x(n-3) + 10 + 5$$
$$= x(n-3) + 15$$

for some k,

$$x(n) = x(n-k) + 5k \quad \text{— } 5$$

$$n-k = 1$$

$n-1 = k$

Equ 5 $x(n) = x(1) + 5(n-1)$

$x(n) = 0 + 5n - 5$

$O(n)$ //

b) $x(n) = 3x(n-1)$ for $n>1$, $x(1)=4$

$x(n) = 3x(n-1)$ — 1

$x(n-1) = 3x(n-1-1) = 3x(n-2)$ — 2

$x(n-2) = 3x(n-3)$ — 3

Sub eq 3 in 2,

$x(n-1) = 3[3x[n-3]]$

$x(n-1) = 9x(n-3)$ — 4

Sub eq 4 in 1,

$x(n) = 3[9x(n-3)]$

$x(n) = 27x(n-3)$

At some k

$x(n) = 3^k x(n-k)$ — 5

$n-k = 1$

$k = n-1$

Eq ⑤ $\Rightarrow x(n) = 3^{n-1} \, a(1)$

$= 3^{n-1} \cdot 4$

$= 3^{n} \cdot 3^{-1} \cdot 4$

$= 3^{n}.$

$\therefore$ The time complexity $= O(3^{n})$

c) $x(n) = x(n/2) + n$ for $n > 1$ $x(1) = 1$
(solve for $n = 2k$)

$x(n) = x(n/2) + c \longrightarrow ①$

$x(n/2) = x(n/4) + c \longrightarrow ②$

$x(n/4) = x(n/8) + c \longrightarrow ③$

Sub ② in ①;

$x(n) = x(n/4) + c + c$

$x(n) = x(n/4) + 2c \longrightarrow ④$

$= x(n/2^2) + 2c$

Sub ③ in ④

$x(n) = x(n/8) + c + 2c$

$x(n) = x(n/2^3) + 3c$

$x(n) = x(n/2^k) + kc$

$n/2^k =$

$n =$

$\log n =$

$\boxed{k = l}$

$n = 2k$ ; $x(1) = 1$

$n = 2k$

$\therefore k = n/2$

$x(n) = x(n/n) + kc$

$x(n) = 1 + kc$

$x(n) = 1 + \log n . c$

Time complexity = $O(\log n)$.

d) $x(n) = x(n/3) + 1$ for $n > 1$ $x(1) = 1$

(solve for $n = 3^k$)

$x(n) = x(n/3) + 1$ —— 1

$x(n/3) = x(n/9) + 1$ —— 2

$x(n/9) = x(n/27) + 1$ —— 3

Sub 2 in 1,

$x(n) = x(n/9) + 2$ —— 4

Sub 3 in 4,

$x(n) = x(n/27) + 3 \longrightarrow$ 5

$= x(n/3^3) + 3$

$x(n) = x(n/3^k) + k$

$x(n) = x(n/3^k) + k$

$= x(n/n) + k$

$= x(1) + k$

$= 1 + k$

$x(n) = \log n$

$\therefore$ Time complexity $= O(\log n)$

2) Evaluate following recurrences complexity.

i) $T(n) = T(n/2) + 1$ where $n = 2^k$ for all $k \geq 0$

$T(n) = T(n/2) + 1$ $\qquad n = 2^k$

Sub $n = 2^k$

$T(2^k) = T(2^k/2) + 1 = T(2^{k-1}) + 1$

$n = k-1$

$T(2^{k-1}) = T\left(\frac{2^{k-1}}{2}\right) + 1 = T(2^{k-2}) + 1$

$n = k-2$

$T(2^{k-2}) = T\left(\frac{2^{k-2}}{2}\right) + 1 = T(2^{k-3}) + 1$

$T(2^1) + T(2^0) + 1$

$n = 2^k \Rightarrow k = \log_2 n$

$T(2^k) = T(2^{k-1}) + 1 = T(2^{k-2}) + 1 + 1 = \dots =$

Since

$2^0 = 1,\ T(2^0) = T(1)$

$T(2^k) = 1 + k$

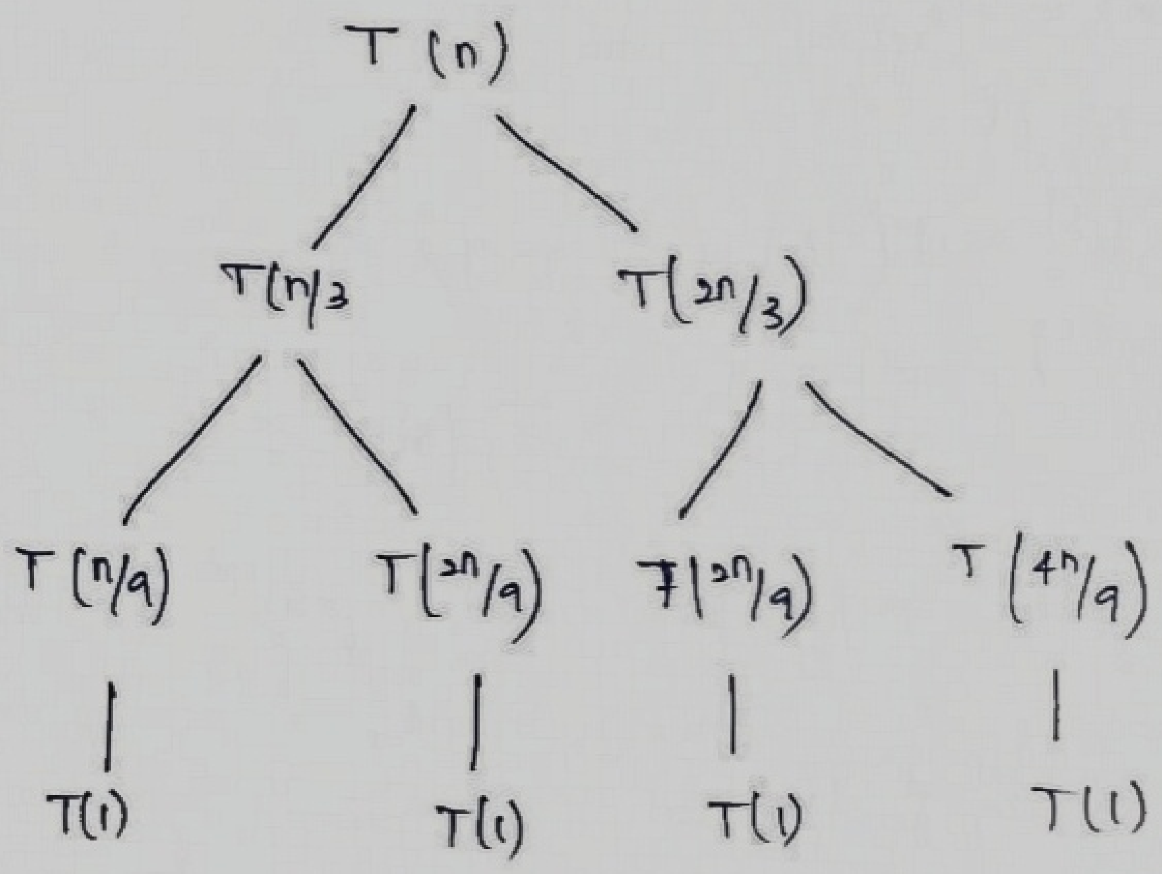
$T(n) = 1 + \log_2 n$

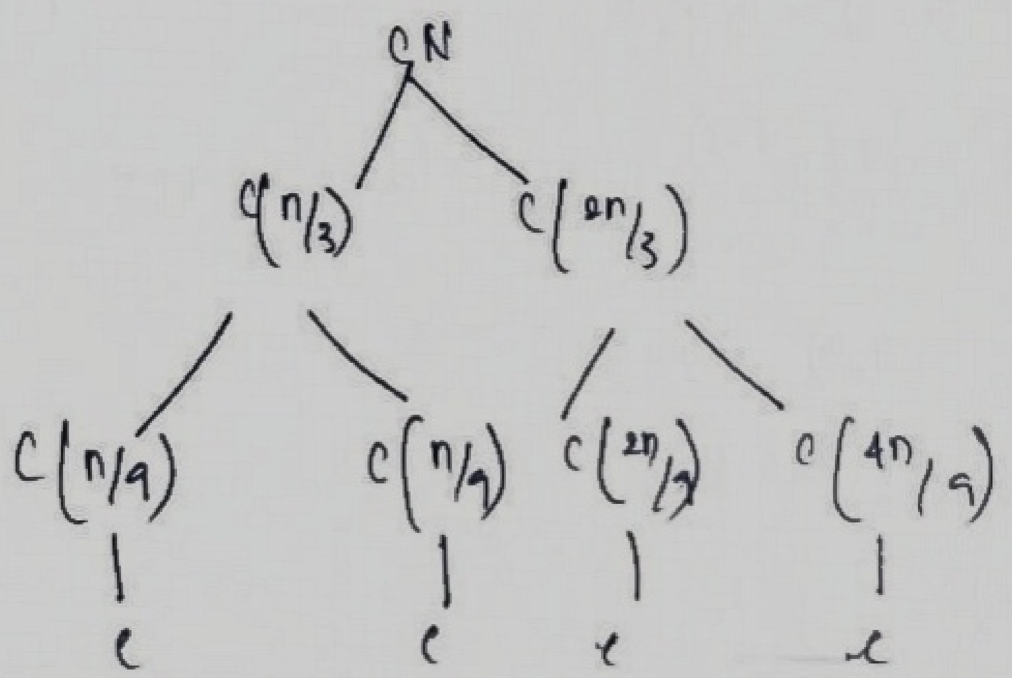
Time Complexity = $O(\log n)$

ii) $T(n) = T(n/3) + T(2n/3) + cn$

We use Recursion tree method

$T(n) = T(n/3) + T(2n/3) + cn$



3) Consider following algorithm

min1(A[0 ... n-1])

if n=1 return A[0] - 1

Else temp = min1 [A [0 ... n-2])

if temp ≤ A[n-1] return temp

else

return A[n-1] - n-1

a) What does this algorithm compute?

This algorithm computes minimum elements in an array A of size n.

If i<n, A[i] is smaller than all elements then, A[j], j = i+1 to n-1, then it returns A[i]. It also returns the leftmost minimal element.

b) Mainly Comparison occurs during recursion and solve it?

So, $T(n) = T(n-1) + 1$ when $n>1$ (one comparison at every step except, $n=1$)

$T(1) = 0$ (No compare when $n=1$).

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

$= 0 + (n-1)$

$= n - 1$

Time complexity : $O(n)$

4 Analyze Order of growth.

1) $F(n) = 2n^2 + 5$ and $g(n) = 7n$ use $\Omega(g(n))$ notation

$F(n) = 2n^2 + 5$ $\qquad$ $F(n) \geq c \cdot g(n)$

$c \cdot g(n) = 7n$

$n = 1$

$F(1) = 2(1)^2 + 5 = 7$

$g(1) = 7$

$n = 2$

$F(2) = 2(2)^2 + 5$

$= 8 + 5 = 13$

$g(2) = 7 \times 2 = 14$

$n = 3$

$F(3) = 2(3$

$= 18 + 5$

$= 23$

$g(3) = 21$

$n = 1, 7 = 7$

$n = 2, 13 = 14$

$n = 3, 23 = 21$

$n \geq 3, F(n) \geq g(n) \cdot c$

F(n) is always greater than or equal to $c \times g($ when, n value is greater or equal to 3.

$\therefore F(n) = \Omega(g(n))$

F(n) is grows more than g(n) from below asymptotically.