R.santhosh
192321141
DAA
CSA-0669
7/06/2023

solve the following recurrence relations:

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

$x(n) = x(n-1)+5 \rightarrow ①$

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

$= x(n-2)+5 \rightarrow ②$

$x(n-2) = x(n-2-1)+5$

$= x(n-3)+5 \rightarrow ③$

sub eq ③ in ①

$x(n-1) = x(n-3)+5+5$

$= x(n-3)+10 \rightarrow ④$

sub eq ④ in eq ①

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

$= x(n-3)+15$

for solve k,

$x(n) = x(n-k)+5k \rightarrow ⑤$

$n-k=1$

$n-1=k$

Eq ⑤

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

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

$O(n)$

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

$x(n) = 3x(n-1) \to ①$

$x(n-1) = 3x(n-1-1) = 3x(n-2) \to ②$

$x(n-2) = 3x(n-3) \to ③$

sub eq ③ in ②,

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

$x(n-1) = 9x(n-3] \to ④$

sub eq ④ in ①,

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

$x[n] = 27x[n-3)$

At some k

$x(n) = 3^k x(n-k) \to ⑤$

$n-k=1$

$k=n-1$

by ⑤ $\Rightarrow x(n) = 3^{n-1} x(1)$

$= 3^{n-1}.4$

$= 3^n . 3^{-1} . 4$

$= 3^n$

∴ The time complexity $= O(3^n)$

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

(solve for $n=2k$)

$x(n) = x(n/2) + c \to ①$

$x(n/2) = x(n/4) + c \to ②$

$x(n/4) = x(n/8) + c \to ③$

sub ② in ①

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

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

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

Sub (3) in (4)

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

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

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

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

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

$x(n) = 1 + kc$

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

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

$x(n) = x(n/3) + 1$ for $n > 1$ $x(1) = 1$ (solve for $n = 3^k$).

$x(n) = x(n/3) + 1 \rightarrow (1)$

$x(n/3) = x(n/9) + 1 \rightarrow (2)$

$x(n/9) = x(n/27) + 1 \rightarrow (3)$

Sub (2) in (1)

$x(n) = x(n/9) + 2 \rightarrow (4)$

Sub (3) in (4).

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

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

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

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

$= T\left(\frac{n}{n}\right) + k$

$= T(1) + k$

$= 1 + k$

$T(n) = \log n$

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

Evaluate following recurrences completely

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 $n = 2^k$

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

$n = k-1$

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

$n = k-2$

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

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

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

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

since

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

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

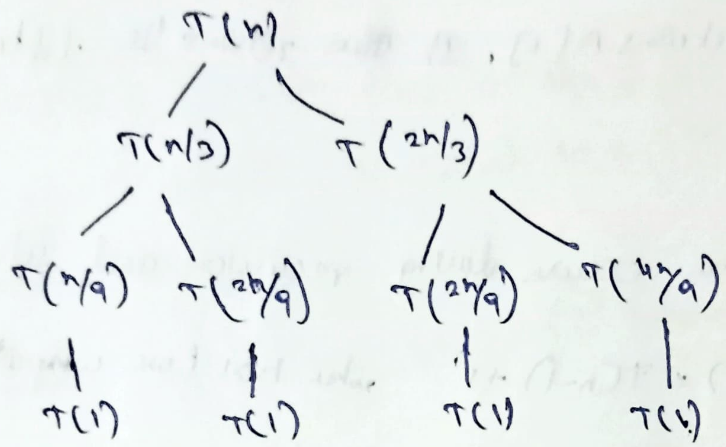
$T(n) = 1 + \log_2 n$

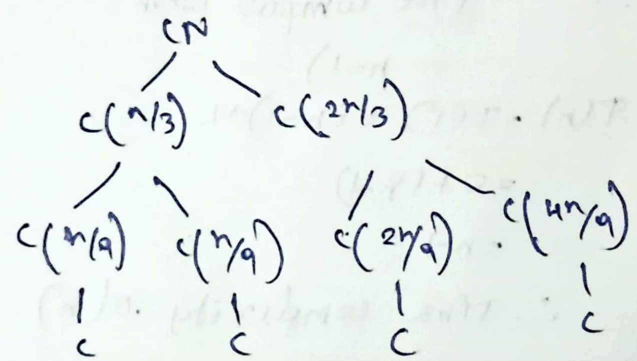
Time complexity $O(\log n)$

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

we Recursion tree method

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





3) consider following algorithm

min 1[A[0 .. n-1]]

if n=1 return A[0] -1

else temp = min 1[A [0.... n-2])

if temp ≤ A[n-1] return temp

else

return A[n-1] -n-1.

a) what does this algorithm compile?

This algorithm compiles minimum elements an array A of size

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

b) manly comparison occurs during recursion and solve it?

so, $T(n) = T(n-1) + 1$ when $n>1$ (one comparison $n=1$)

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

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

$= 0 + (n-1)$

$= n-1$

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

4) Analyze order of growth.

1) $f(n) = 2n^2 + 5$ and $g(n) = 7n$ we

$f(n) = 2n^2 + 5$ $\qquad$ $f(n) \geq c\,g(n)$

$c.g(n) = 7n$

$n=1$

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

$g(1) = 7$

$n=2$

$f(2) = 2(2)^2 + h$

$= 8 + 5 = 13$

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

$n=3$

$f(3) = 2(2)^3 + 5$

$= 23$

$g(3) = 21$

$n=1$, $7=7$

$n=2$, $13=14$

$n=3$, $23=21$

$n \geq 3, f(n) \geq g(n).c$

$f(n)$ is always greater than or equal to $c \geq c$
where, 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.