Aravind . S
142321081
CSA 0669

Assignment :

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

$n(1) = 0$

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

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

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

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

$n(n-2) = n(n-3)+5$

$n(n-2) = n(n-3)+5$ - (3)

sub eq (3) in (2)

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

$n(n-1) = n(n-3)+10$ - (4)

sub (4) in (1)

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

$= n(n-3)+15$

$= n(n) = n(n-K)+5K$ - (5)

$n-K = 1$

$n-1 = K$

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

$\Rightarrow O(n)$

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

$n(n) = 3n(n-1)$

$n(1) = 4$

$n(n) = 3n(n-1)$ - 1

$n(n-1) = 3n(n-1-1)$

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

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

Sub 3 in 2

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

$n(n-1) = 9n(n-3)$ - 4

Sub 4 in 1

$n(n) = 3n(9n(n-3))$

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

$n(n) = 27n(n-k)$

$3^{n-1} \cdot 4 = O(3^n) = 3^k n(n-k)$ - 5

$n-k = 1 \Rightarrow k^2 n^{-1}$

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

c) $n(n) = n(n/2) + n$ $\quad n>1 \quad (n = 2K)$

$n(1) = 1$

$n(n) = n(n/2) + n$ - ①

$n(n/2) = n(n/4) + n$ - ②

$n(n/4) = n(n/8) + n$ - ③

Sub ③ in ②

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

$n(n/2) = n(n/8) + 2n$ - ④

Sub ④ in ①

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

$n(n) = n(n/8) + 3n$ - ⑤

$n(n) = (n/2) + n + (n/4) + n \; \cdots \; (n/2k) + n^k$

$m(n) = \frac{n}{2^k} + kn$

$\frac{n}{2k} = 1$

$n = 2k$

$\log n = \log 2k$

$\log n = k \log 2$

$\log n = k$

$O(\log n)$

d) $n(n) = n(n/3) + 1$

$n(n) = n(n/3) + 1$ — (1)

$n(n/3) = n(n/9) + 1$ — (2)

$n(n/9) = n(n/27) + 1$ — (3)

Sub (3) in (2)

$n(n/3) = n(n/27) + 2$ — (4)

Sub (4) in (1)

$n(n) = n(n/27) + 3$ — (5)

$n(n) = n + \frac{n+1}{3} + \frac{n+2}{3} + \frac{n+3}{27} \dots \frac{n}{3^k} + k$

$\frac{n}{3^k} = 1$

$n = 3^k$

$\log n = \log 3^k$

$k = \log n$

$\therefore O(\log n)$

Scanned with OKEN Scanner

2) Evaluate the following

(i) $T(n) = T(n/2) + 1$ where $n = 2^K$ for all $K > 0$

$T(n) = aT(n/b) + f(n)$

$= f_n = n^K \log_n^P$

where $K >= 0$ and it is a real number

case 1:

$\log_a^b > K$ then $\Theta(n \log_a^b)$

case 2:

$\log_a^b = K$

$P > -1$ $\Theta(n^K \log n^{P+1})$

$P = -1$ $\Theta(n^K \log \log n)$

$P < -1$ $\Theta(n^K)$

case 3:

$\log_a b < K$

$P \geq 0$ $\Theta(n^K \log_n^P)$

$P \lneq 0$ $\Theta(n^K)$

STEP 1: $T(n) = T(n/2) + 1$

$T(n) = aT(n/b) + f(n)$

$a = 1, b = 2$

$\Rightarrow \log_a^b$

$= \log 2$

STEP 2 : $F(n) = 1$

$f(n) = n^k \log^p n$

$p = 1$

$k = 1$

$\log_b^a < k$

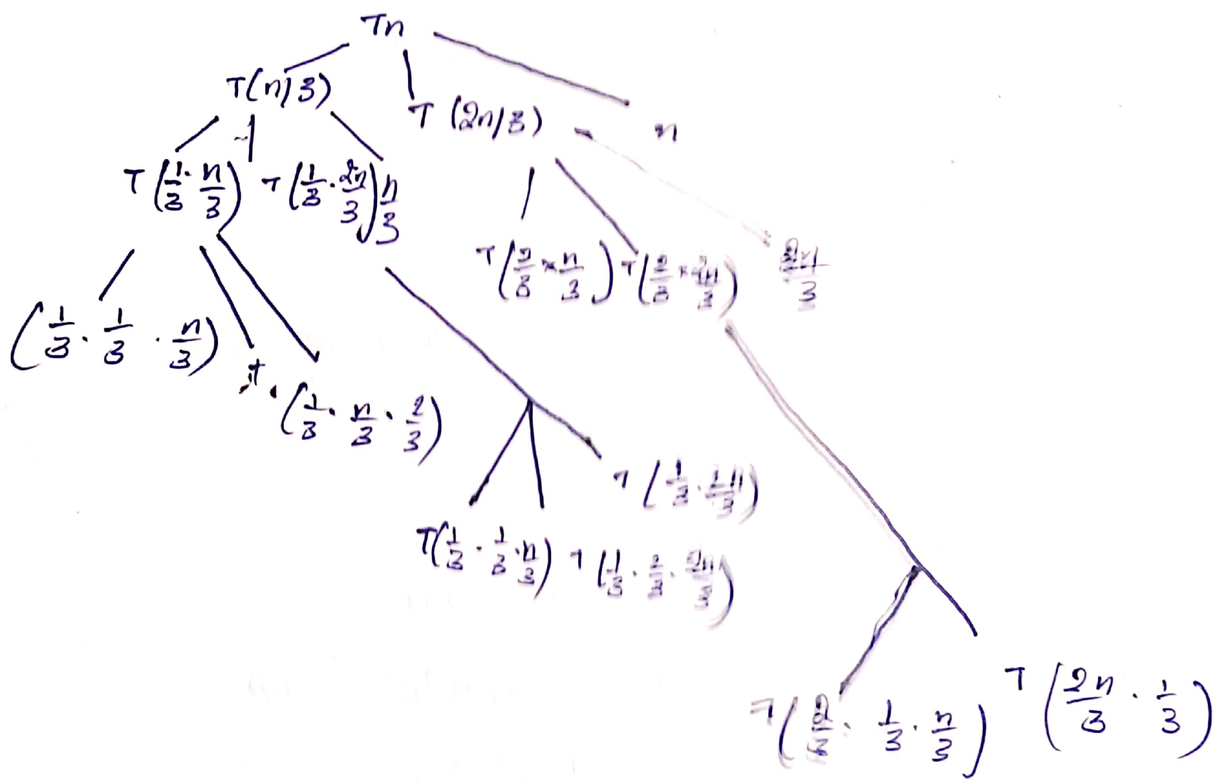
STEP 3 : $\Theta(n^k \log^p n)$

$= \Theta(n^1 \log^1 n)$

$= \Theta(n \log^1 n)$

Time complexity $= \Theta(\log n)$

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



assume $\frac{2^i}{3^k} = 1$

$\Rightarrow \frac{n}{(\frac{3}{2})^k} = 1$

$\Rightarrow n = (\frac{3}{2})^k$

$\log n = k \log (\frac{3}{2})$

$k = \frac{\log n}{\log \frac{3}{2}}$

Time complexity $= \Theta(n^k)$

$\Rightarrow \Theta(n^{\log_{\frac{3}{2}} n})$

3) Min, (A[0 . . . n-1])

if n=1

return A[0]

else :

temp = Min, [A(0 : : : n-2])

If temp ≤ = A(n-1)

return temp

else:

return A (n-1)

This algorithm computes minimum element in an Array A of size n.

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