1. Solve the following recurrence relations

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

$x(n) = x(n-1) + 5$ ---- ①

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

$= x(n-2) + 5$ ---- ②

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

$= x(n-3) + 5$ ---- ③

Sub eq ③ in ②

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

$= x(n-3) + 10$ ---- ④

Sub eq ④ in eq ①

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

$= x(n-3) + 15$

for some k,

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

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

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

At some k

$x(n) = 3^k x(n-k)$ ---- 5

$n-k = 1$

$k = n-1$

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

$= 3^{n-1} 4$

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

$= 3n$

$\therefore$ 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

$x(n) = x(n/8) \cancel{*} c + 2c$

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

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

$2^k =$

$n = 2$

$\log_2 n =$

$\boxed{k =}$

$n; k$

$\therefore k = n/2$

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

$x(n) = x(1/x) + kc$

$x(n) = 1 + kc$

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

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

$x(n) = x(n/27) + 3$ —— ⑤

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

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

$= x(n/n) + k$

$= x(1) + k$

$= 1 + k$

$x(n) = \log n$

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

2) Evaluate following recurrences Completely

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

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

Sub $n = 2^k$

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

$n = k-1$

$T(2k-1) = T[\frac{2 \cdot k-1}{2}) + 1 = T(2^{k-2}] + 1$

$n = k-2$

$T[2^{k-2}) = T[\frac{2^{k-2}}{2}] + 1 = T[2^{k-3}] + 1$

$T[2^1] + T[2^8] + 1$

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

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

Since

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

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

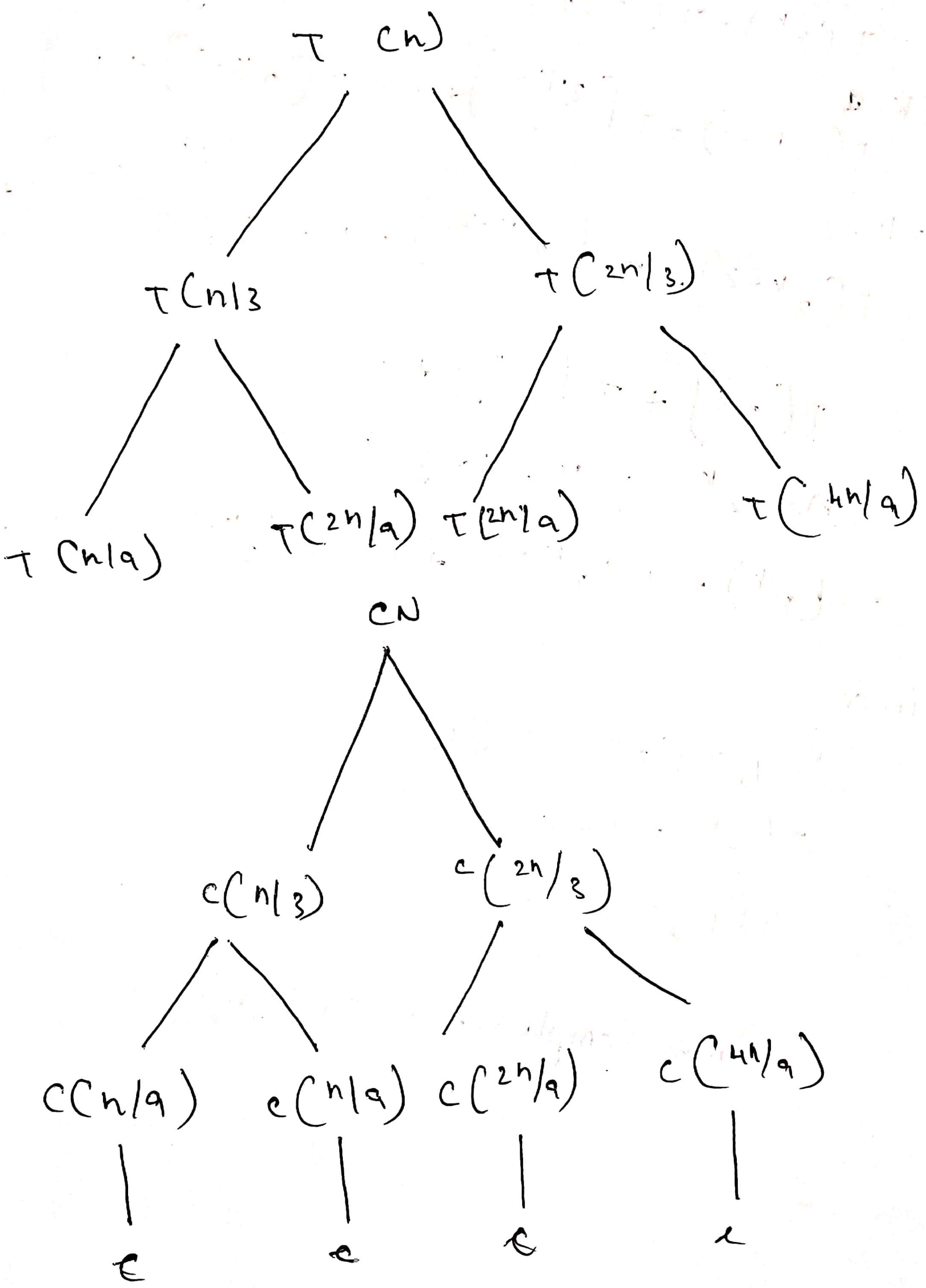
$T(n) = 1 + \log_2 n$

Time Complexity $- O(\log n)$

iii) $T(n) = T(n/3), T[2n/3] + cn$

We use Recursion tree method

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



3) Consider following algorithm

min 1 $[A[0 \ldots n-1]]$

if $n=1$ return $A[0]-1$

Else temp = min 1 $[A[0 \ldots n-2]]$

if temp $\leq A[n-1]$ return temp

else

return $A[n-1]-n-1$

a) What does this algorithm Compete ?

This algorithm computes minimum elements is an aaray 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 $A[i]$.

If also returns the leftmost minimal element

b) Mainly comparion occurs during recursion and solve it;

(one comparision an every step except, $n=1$)

So, $T(n) = T(n-1)+1$ when $n>1$

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

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

$= 0 + (n-1)$

$= n-1$

$\therefore$ Time Complexity $= O(n)$

Scanned with OKEN Scanner

4) Analyze Order of growth

1) $F(n) = 2n^2+5$ and $g(n) = 7n$ use - a $[g(n)]^2$

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

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

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

$n = 1$

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

$g(1) = 7$

$n = 2$

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

$= 8 + 5 = 13$

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

$n = 3$

$F(3) =$

$=$

$= 23$

$g(3) = 21$

$n = 1, 7 = 7$

$n = 2, 13 = 14$

$n = 3, 23 = 21$

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

F(n) is always greater than or equal to Cxg to 3 when, n value is greater or equal;

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

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