NAME: RIDA . S

REG NO: 192321010

SUB: CSA 0669

CODE: DAA

DATE: 06/06/2024

1) Solve the following recurrence relation:

$x(n) = x(n-1) + 5 \longrightarrow ①$

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

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

$x(n-2) = x(n-3) + 5 \longrightarrow ③$

Sub Eq ③ in ②:

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

$x(n-1) = x(n-3) + 10 \longrightarrow ④$

Sub Eq ④ in ①;

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

$x(n) = x(n-3) + 15 \longrightarrow ⑤$

For some k;

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

$\cancel{x(n)=}$ $n-k=1; n-1=k$.

Eq ⑤ $\rightarrow x(n) = x(1) + 5(n-1) = 0 + 5n - 5$

$x(n) = 5n - 5$

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

2) $x(n) = 3x(n-2)$ for $n > 1$, $x(1) = 4$:

$x(n) = 3x(n-1) \rightarrow ①$; $x(n-1) = 3x(n-1-1) = 3x(n-2) \longrightarrow ②$

$x(n-2) = 3x(n-3) \longrightarrow ③$

Sub Eq ③ in ②;

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

$x(n-1) = 9x(n-3) \longrightarrow ④$

Sub Eq ④ in ①:

$x(n) = 3[9x(n-2)]$

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

At some k;

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

$n-k=1$

$k=n-1$

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

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

$= 3^n$

Time Complexity $= O(3^n)$

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

$x(n) = x(n/2) + c$ — (1)

$x(n/2) = x(n/4) + c$ — (2)

$x(n/4) = x(n/8) + c$ — (3)

Sub (2) in (1);

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

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

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

Sub (3) in (4): $x(n) = x(n/8) + c + 2c$

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

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

$\boxed{n=2^k}$; $\boxed{x(1)=1}$.

$x(n) = x(\cancel{n}/\cancel{n}) + kc$

$x(n) = 1 + kc$

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

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

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

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

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

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

Sub ② in ①: $x(n) = x(n/9) + 2 \rightarrow ④$

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

Sub ③ in ④: $x(n) = x(n/3^3) + 3 \rightarrow ⑤$

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

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

$n/3^k = 1$	$x(n) = x(n/3^k) + k$
$n = 3^k$	$= x(n/n) + k$
$\log_3 n = \log_3 3^k$	$= x(1) + k$
$\boxed{k = \log_3 n}$	$= 1 + k$
	$x(n) = 1 + \log_3 n$

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

2) Evaluate following recurrence completely:

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

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

Sub $n = 2^k$

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

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

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

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

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

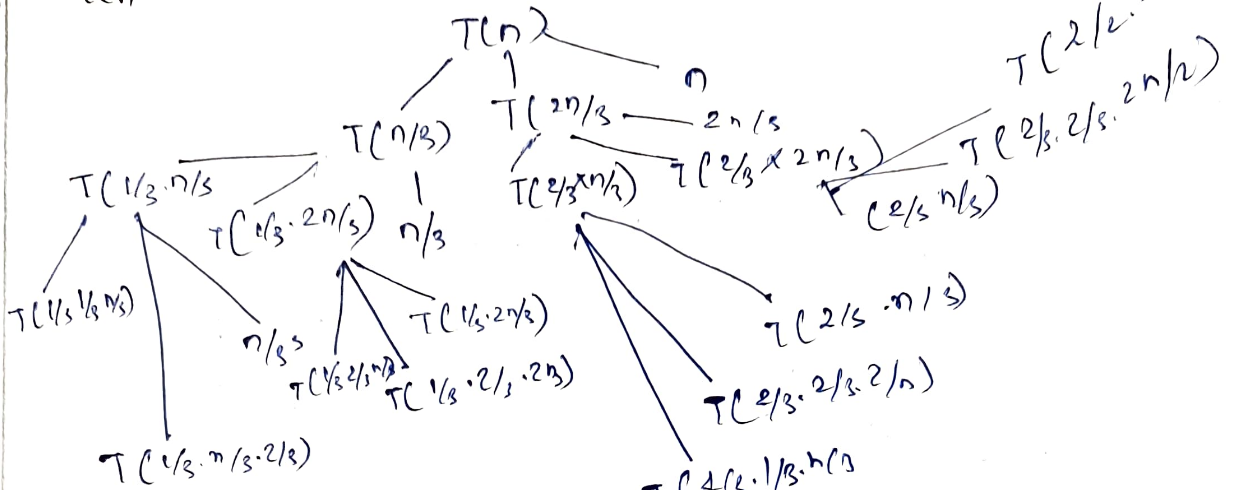
Since,

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

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

$\boxed{T(n) = 1 + \log_2 n}$. Time complexity $\} = O(\log n)$

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



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

$= n / (3/2)^k = 1$

$n = (3/2)^k$

$\log n = k \log(3/2)$

$k = \log n / \log 3/2$

$k = \log 3/2$

$\boxed{k = \log^n 3/2}$

Time complexity $= O(nk)$

Time complexity $= O(n \log^n 3/2)$

3) Consider the given Algorithm:

a) This algorithm computes minimum element 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) $T(n) = T(n-1) + 1$, when $n > 1$

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

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

$= 0(n-1)$

$= n-1$

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

Analyze order of Growth:

(i) $f(n) = 2n^2+5$ and $g(n) = 7n$ use the $\Omega(g(n))$ notation

Given,

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

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

$n = 1$

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

$g(1) = 7$

$n = 1$, $7 = 7$

$n = 2$, $13 = 14$

$n = 3$, $23 = 21$

$$\boxed{f(n) \geq c \cdot g(n)}$$

$n=2$:

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

$= 8 + 5 = 13$

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

$n=3$:

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

$= 18 + 5 = 23$

$g(3) = 21$

$f(n) \geq c \cdot g(n)$ where, n value is $\geq$ to 3.

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

$\Rightarrow f(n)$ is more than $g(n)$ from asymptotically