CSA0669 - Design and Analysis of Algorithm.

Analytical Question. Assignment

By
R. Jagan
192321102
B.Tech IT.

1. Solve the following recurrence relation.

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

$x(1) = 0$

$n = 2$

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

$= x(1)+5$

$= 0+5$

$\boxed{x(2) = 5}$

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

$= x(2)+5$

$= 5+5$

$\boxed{x(3) = 10}$

$n = 4$

$x(4) = x(4-1)+5$

$= x(3)+5$

$\boxed{x(4) = 15}$

$\therefore$ For this recurrence relation, each time is 5 is more than previous time

$\boxed{x(n) = 5n-5 \text{ for } n>1}$

b) $x(n) = 3x(n-1)$ for $n > 1$ $\boxed{x(1) = 4}$

$x(1) = 4$

$n = 2$

$$x(2) = 3x(2-1)$$
$$= 3x(1)$$
$$= 3(4)$$
$$\boxed{x(2) = 12}$$

$n = 3$

$$x(3) = 3x(3-1)$$
$$= 3x(2)$$
$$= 3(12)$$
$$\boxed{x(3) = 36}$$

$n = 4$

$$x(4) = 3x(4-1)$$
$$= 3x(3)$$
$$= 3(36)$$
$$\boxed{x(4) = 108}$$

For this recurrance relation, each term is 3 times the previous term.

$\boxed{\text{So, } x(n) = 4 * 3^{n-1} \text{ for } n > 1}$

c)

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

$x(1) = 1$

$n = 2$

$n = 2^1 = 2$

$$x(2) = x\left(\frac{2}{2}\right) + 2$$
$$= x(1) + 2$$
$$= 1 + 2$$
$$\boxed{x(2) = 3}$$

$n = 4$

$n = 2^2 = 4$

$$x(4) = x(4/2) + 4$$
$$= x(2) + 4$$
$$= 3 + 4$$
$$\boxed{x(4) = 7}$$

$n = 8$

$n = 2^3 = 8$

$$x(8) = x(8/2) + 8$$
$$= x(4) + 8$$
$$= 7 + 8$$
$$\boxed{x(8) = 15}$$

For this recursion relation, $2^n - 1 = 2^k - 1$

for $\boxed{n = 2^K}$, $\boxed{x(2K) = 2^{2K} - 1}$.

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

$x(1) = 1$

$x(1) = 1$

$n = 3^k$

$n = 3$

$n = 3^1 = 3$

$x(3) = x(3/3) + 1$

$n = 3^2 = 9$

$= x(1) + 1$

$n = 3^3 = 27$

$= 1 + 1$

$\boxed{x(3) = 2}$

$n = 9$

$x(9) = x(9/3) + 1$

$= x(3) + 1$

$= 2 + 1$

$\boxed{x(9) = 3}$

$n = 27$

$x(27) = x(27/3) + 1$

$= x(9) + 1$

$= 3 + 1$

$\boxed{x(27) = 4}$

for this recurrence relation,

$x(n) = \log_3 n$

$\boxed{x(r) = \log_3 3K}$

2. Evaluate the following recurrence completely

(i) $T(n) = T(n/2)+1$, where $n = 2k$ for all $k > 0$.

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

Since

$n = 2^k$, we can rewrite $n/2$ as 2^{k-1}

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

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

$$T(2^k) = T(2^{k-3})+1+1+1 \ldots$$

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

Since $T(1)$ is constant, lets call it 'a'

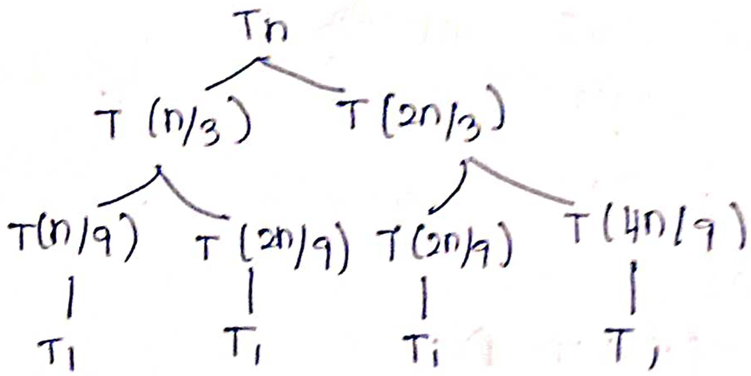
$$T(2^k) = a+k$$

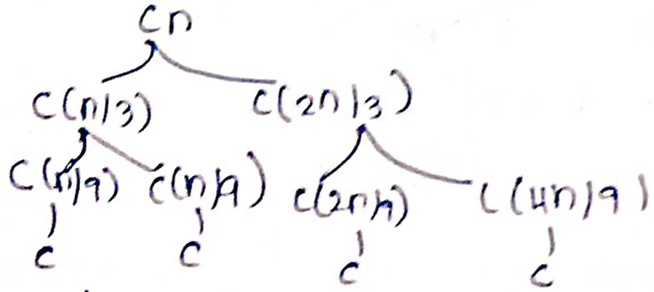
$$\therefore T(n) = a+\log_2(n)$$

---

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

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





length = $\log_3 n$ (divided by 3)

$$T(n) = cn \log_3 n$$

$$\Rightarrow \omega(n \log n).$$

3. consider following algorithm.

min [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 compute.

The algorith compute minimum element in an array n 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 left most minimal element.

b) Set up a recursive relation for algorithms basic operations count and solve it.

main by comparision occurs during recursion

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

$T(1) = 0$

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

$= 0 + (n-1)$

$= n-1$

∴ Time Complexity = $O(n)$

4. Analyze the order of growth

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

$n = 1$

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

$g(1) = 7$

$n = 2$

$F(2) = 2(2)^2 + 5 = 13$

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

$n = 3$

$F(3) = 2(3)^2 + 5 = 23$

$g(3) = 21$

$n = 1, \ 7 = 7$

$n = 2, \ 13 = 14$

$n = 3, \ 23 = 21$

$n \geq 3$

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

F (n) is always greater then or equal to c · g (n) when n value is greater or equal to 3

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

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