Analytical Assignment - 1

CSA - 0672

Name: Sandhiya. S
RegNo: 192321015
Course Code : CSA 0672

1) Solve the following recurrence relations

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

Given:

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

$n = 1$ $\quad n(1) = 0$

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

$$= x(1) + 5$$

$$= 0 + 5 = 5 \rightarrow (1)$$

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

$$= x(2) + 5$$

$$= 10 \rightarrow (2)$$

$n = 4$ $\quad x(4) = x(4-1) + 5$

$$= x(3) + 5$$

$$= 15$$

The given equation is $x(n) = x(1) + (n-1)d$ in the equation $d = 5$ and $x(1) = 0$

$$x(n) = 0 + 5(n-1)$$

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

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

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

Given:

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

$x(1) = 4$

Sub $n=2$ | Sub $n=3$ | Sub $n=4$

$x(2) = 3x(n-1)$
$= 3x(2-1)$
$= 3x(1)$
$= 3\times 4$
$= 12$

$x(3) = 3x(3-1)$
$= 3x(2)$
$= 3\times 12$
$= 36$

$x(4) = 3x(4-1)$
$= 3x(3)$
$= 3\times 36$
$= 108$

The general form of equation is $x(n) = 3^{n-1}.\, x(1)$

$x(n) = 3^{n-1}.\, 4$

$x(n) = 3^{n-1}.\, 4$ is the recurrence relation

c) $x(n) = x(n/2) + n$ for $n=1$, $x(1)=1$ (solve for $n=2k$)

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

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

$x(2k) = x\left(\frac{2k}{2}\right) + 2k$

$x(2k) = xk + 2k$

Sub $k=1$

$x(2-1) = x(1) + 2 = 2.1 = 1+2$
$= 3$

Sub $k=2$

$x(2-2) = x(2) + 2..2$

$x(2) = x(1) + 2 = 1 + 2 = 3$

$x(4) = x(2) + 4$

$= 3 + 4 = 7$

Sub $k = 3$

$x(2 \cdot 3) = x(3) + 2 \cdot 3$

$x(3) = x(1.5) + 3$

The general equation for given equation is

$x(2k) = x(k) + 2k$

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

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

$x(1) = 1 \quad n = 3k$

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

$x(3k) = xk + 1$

Sub $k = 1$

$x(3 \cdot 1) = x(1) + 1$

$x(3) = 1 + 1 = 2$

Sub $k = 2$

$x(3 \cdot 2) = x(2) + 1$

$x(6) = x(2/3) + 1$

Sub $k = 3$

$x(3 \cdot 3) = x(6) + 1$

$= 2 + 1$

$x(9) = 3$

The general equation for given expression is

$x(3k) = 1 + \log_3(k)$

Evaluate the following recurrence completely:

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

$n = 2k \quad k = \log n$

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

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

$T(2.k) = T(k/2) + 2$ (if k is even)

$T(2.k) = T\left(\frac{(k-1)}{2}\right) + 2$ (if k is odd)

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

Recurrences $\Rightarrow T(n) = \theta(\log n)$

ii) $T(n) = T(n/3) + T\left(\frac{2n}{3}\right) + cn$ where c is a constant and 'n' is the input size

$T(n) = aT\left(\frac{n}{6}\right) + f(n)$

$a = 2, \quad b = 3 \quad f(n) = cn$

Master theorem

$f(n) = \theta(n^c)$ where $c < \log_b a$, then $T(n) = \theta(n \log_b a)$

$f(n) = \theta(n \log_b a)$ then $T(n) = \theta(n \log_b a \log x)$

$f(n) = \Omega(n^c)$ where $c > \log_b a$, as $(a/b) \leq \log(n)$

for $K < 1$

$T(n) = \theta(f(x))$

find $\log_b a = \log_b a = \log_3 2$

$f(n) = kn = n \log_b a$

Recurrence Relation $T(n) = \theta(n)$

$= a(2) + 2 .. k$

The following recursion algorithm.

min 1 (A [0 .... n-1]) - 1
if n=1 return A[0] - 1
else

temp = min (A [0 ... n-2])
if temp <= A(n-1) return temp
else

Return A(n-1)

a) What does the algorithm compute

This algorithm computes the minimum element in an array A size n using a recursive approach.

Base case:

If the array has only one element (n ==1) it return that single element as the minimum

Recursive case:

+ If the array has more than one element (n=1) the function makes a recursive call to find the min element in subarray consisting of the first (n-1) element.

+ The function return the smaller of these two values.

b) Set up a recursive relation for the algorithm basic operation and solve it.

Min1 (A [0... n-1])
if n=1
return A[0]
else

temp = Min ((A [0 ... n-2]) - n-1

if temp <= A (n-1)

return temp

else

Return A [n-1]

$T(n)$ = No. of basic operation

if $n=1$ then $T(1) = 0$

$T(n) = T(n-1) + 1$ is the recurrence

$T(1) = 0$

$T(2) = T(2-1) + 1$

$= T(1) + 1$

$= 0 + 1$

$T(2) = 1$

$T(3) = T(3-1) + 1$

$= T(2) + 1$

$= 1 + 1$

$= 2$

$T(4) = T(4-1) + 1$

$= T(3) + 1$

$= 2 + 1$

$= 3$

$T(n) = n - 1.$

Time Complexity $= O(n)$

4) Analyze the order of growth.

i) $F(n) = 2n^2 + 5$ Use the Ω (g(n)) notations

$g(n) = 2n$

$n) = 2n^2 + 5$

$g(n) = 7n$

if $n=1$ $\quad F(n) = 2(1)^2 + 5$

$= 7$

$g(n) = 7(1)$

$= 7$

$n=2 \Rightarrow F(n) = 2(2)^2 + 5$ $\qquad g(n) = 7(2)$

$= 13$ $\qquad = 14$

$n=3 \Rightarrow F(n) = 2(3)^2 + 5$ $\qquad g(n) = 7(3)$

$= 23$ $\qquad = 21$

$n=4 \Rightarrow F(n) = 2(4^2) + 5$

$= 2(16) + 5$

$= 37.$

$F(n) \geq g(n) \cdot c$ condition satisfies at $n=1$ onwards so

$\Omega(T(n))$ is the occurence relation

Time complexity is $\Omega(n)$