ASSIGNMENT ⇒ I

Name of the Student : A. Manoj Reddy

Register No. : 192311171

Course Code : CSA0670

Name of Subject : Design and Analysis of algorithm

Date of Submission :

(1) Solve the following recurrence relation.

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

(A) Step1: write down the first two terms to identify the pattern

$x(1)=0$

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

$x(3) = (x(2))+5 = 10$

$x(4) = x(3)+5 = 15$

Step 2: Identify the pattern (or) the general term

→ The first term $x(1)=0$

The common difference $d=5$

The general Formula for the n^{th} term of an AP is

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

Substituting the given values

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

The solution is $x(n) = 5(n-1)$

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

(A) Step1: write down the first two terms to identify the pattern

$x(1) = 4$

$x(2) = 3x(1) = 3\cdot4=12$

$x(3) = 3x(2) = 24$

$x(4) = 3x(3) = 36$

Step2:- identify the general term

$$x(n) = x(1)\cdot r^{n-1}$$

Substituting the given values

$$x(n) = 4\cdot3^{n-1}$$

The solution is $x(n) = 4\cdot3^{n-1}$

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

For $n = 2^k$, we can write recurrence interms of k

1. Substitute $n = 2^k$ in the recurrence

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

2. write down the first few terms to identify the pattern

$x(1) = 1$

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

$x(4) = x(2^2) = x(2) + 4 = 3 + 4 = 7$

$x(8) = x(2^3) = x(4) + 8 = 7 + 8 = 15$

3. identify the general term by finding the pattern we

Observe that $x(2^k) = x(2^{k-1}) + 2^k$

we sum the series

$$x(2^k) = 2^k + 2^{k-1} + 2^{k-2} + \cdots$$

$$S = a \frac{r^n - 1}{r - 1}$$

Here $a = 2$, $r = 2$ and $n = k$

$$S = 2 \frac{2^k - 1}{2 - 1} = 2(2^k - 1) = 2^{k+1}$$

adding the +1 term

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

Sol is

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

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

For $n = 3^k$, we can write the recurrence interms of k

i) Substitute $n = 3^k$ in the recurrence $x(3^k) = x(3^{k-1}) + 1$

2. Write down the first few terms to identify the pattern

$x(1)=1$

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

$x(9) = x(3^2)+1 = (x(3)+1 = 2+1 = 3$

$x(27) = x(3^3) = x(9)+1 = 3+1 = 4$

3. identify the general term:

we observe that

$x(3^k) = x(3^{k-1})+1$

Sum up the series

$x(3^k) = 1+1+1+\cdots$

$x(3^k) = k+1$

The solution is $x(3^k) = k+1$

2. Evaluate the following recurrence complexity

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

The recurrence relation can be solved using iteration method

1) Substitute $n=2^k$ in the recurrence

2) iterate the recurrence

for $k=0$: $T(2^0) = T(1) = T(1)$

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

$k=2$: $T(2^2) = T(4) = T(2)+1 = T(1)+1+1 = T(1)+2$

$k=3$: $T(2^3) = T(8) = T(4)+1 = T(1)+2 = T(1)+3$

3) generalize the pattern

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

Since $n=2^k$, $k=\log n$

$T(n) = T(2^k) = T(1)+\log_2 n$

4) $T(n) = O(\log n)$

ii) $T(n) = T(n/3) + (2n/3) + Cn$ where C is constant and n is input size

The recurrence can be solved using the master's theorem for divide and conquer recurrence of the form

where $a=2$, $b=3$, and $F(n) = Cn$

lets determine the value of $\log_b a$

$$\log_b^a = \log 2$$

using the properties of logarithm

$$\log_3 2 = \frac{\log 2}{\log 3}$$

now we compare $F(n) = Cn$ with $n \log_3 2$

$$F(n) = O(n)$$
$$n = n^1$$

since $\log_3 2$ we are in the third case of the master's theorem $F(n) = O(n^c)$ with $C > \log_b a$

The solution is : $T(n) = O(F(n)) = O(C(n)) = O(n)$

3. consider the following recurrence algorithm?

min [A[0 ...n-2)]

if n=1 return A[0]

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

if temp <= A(n-1) return temp

else return A[n-1]

a) what does this algorithm compute?

min [A[0, ...n-2]) computer the minimum value in the array 'A' from index '0' for 'n-1', if does if does this value in the array A from index '0' for 'n-1' if does this by recursively finding the minimum value in the sub array $A[0 \ldots n-2]$ and then comparing if with the last element $A[n-1]$ to determine the overall maximum value.

b) Setup a recurrence relation for the algorithm basic operation count and solve:

The solution is

$T(n) = n$

This means the algorithm perform n basic operation for an input array of size n.

4. Analyze the order of growth.

i) $F(n) = 2n^2 + 5$ and $g(n) = 7n$ use the Ω $g(n)$ notation.

To analyze the order of growth and use the Ω notation, we need to compare the given function $F(n)$ and $g(n)$ given function

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

$g(n) = 7n$

The notation $\Omega(g(n))$ notation describes a lower bound on the growth rate that for sufficiency large n, $F(n)$, grows at least as far as $g(n)$

$F(n) = C \cdot g(n)$

lets analyze $F(n) = 2n^2 + 5$ with to $g(n) = 7n$

1) identify Dominant terms:

→ The dominant term in $F(n)$ is $2n^2$ since it grows faster then the constant term as n increases

∴ The dominant term in $g(n)$ is $7n$

2) establish the inequality

→ we want to find constant and no such that:

$$2n^2 + 5 \geq c \cdot 7n \text{ for all } n \geq n_0$$

3) simplify the inequality

→ we want to find constant and no such that

$$2n^2 \geq 7cn$$

→ divide both sides by n

$$2n \geq 7c$$

→ Solve for n:

$$n \geq 7c/2$$

4) choose constants

let $c = 1$

$$n \geq \frac{7 \cdot 1}{2} = 3.5$$

∴ For $n \geq n_1$ the inequality holds.

$2n^2 + 5 \geq 7n$ for all $n \geq n$

$$2n^2 + 5 \geq 7n$$

$F(n) = 2n^2+5 = \Omega$ (7n)

$F(n) = \Omega(n^2)$ f(n) faster than $7n$.