NAME :- V. Sreekanth.
192372037.

1. Solve the following recurrence relation.

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

(1) write down the first two terms to identify the pattern

$x(1) = 0$

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

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

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

(2) Identify the pattern (or) the general term.

⇒ The first term $x(1) = 0$

The common difference $d = 5.$

The general formula for the nth term of an ap is.

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

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 $x(1) = 4$

(1) write down the first two terms to identify the pattern

$x(1) = 4$

$x(2) = 3x(1) = 3.4 = 12.$

$x(3) = 3x(2) = 36$

$x(4) = 3x(3) = 108.$

Substituting the given values

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

The solution is

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

(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 in terms 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.

$x(1) = 1$

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

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

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

(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:

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

Since $x(1) = 1$!

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

The geometric series with the term $a = 2$ and the last term 2^k except for the additional $+1$ term.

The sum of a geometric series with ratio $r = 2$ is given by

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

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

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

Solution 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 in term of k

(1) 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

$$x(1) = 1$$

$$x(3) = x(3^{1}) = x(1)+1 = 1+1 = 2.$$

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

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

(3) Identify the general term:

we observe that:

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

Summing up the series

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

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

The Solution is

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

(a) (ii) Evaluate the following recurrences complexity.

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

The recursence relation can be solved using iteration method.

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

(2) Iterate the recurrence.

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

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

$k=2 = T(2^2) = T(4) = T(2)+1 \; (T(1)+2)+1 = T(1)+2.$

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

(3) generalize the pattern

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

Since $n=2^k$, $k = \log_2 n$

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

(4) Assume $T(1)$ is a constant c.

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

The solution is

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

(ii) $T(n) = T(n/3) + T(2n/3) + Cn$ 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

$T(n) = aT(n/b) + f(n)$

where $a = 2$, $b = 3$ and $f(n) = cn$.

Lets. determine the value of $\log_b a$

$\log_b a = \log_3 2$

using the properties of algorithms

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

Now we compare $f(n) = cn$ with $n^{\log_3 2}$:

$f(n) = \Theta(n)$

$n = n^1$.

Since. $\log_3 2$ we are in the third one case of the master's theorem

$f(n) = O(n^c)$ with $c > \log_b a$.

The solution is:

$T(n) = O(f(n)) = O(cn) = O(n)$.

(3) Consider the following recurrence algorithms.

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

if n = 1 return A[0]

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

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

else

return A[n-1]

(a) what does this algorithm compute?

The given algorithm, $\min(A[0,\dots n-1])$ computes the minimum value in the array 'A'. from index '0' for (n-1). in the sub array $A[0\dots n-2]$ and then Comparing it 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 it.

The Solution is

$T(n) = n$

This means the algorithm performs n basic operations for an input array. of size n.

(4) Analyse the order of growth.

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

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

$g(n) = 7n$

Order of growth using $\Omega(g(n))$ Notation.

The notation $\Omega(g(n))$ describes a lower bound on the growth rate that for sufficiently large n, $f(n)$, grows atleast as $f(x)$ as $g(n)$.

$f(n) \geq (g(n))$

Let's analyze $f(n) = 2n^2 + 5$ with respect to $g(n) = 7n$

(1) identify Dominant terms:

→ the dominant terms in $f(n)$ is $2n^2$ since its grows faster then the constant terms as n increases.

→ the dominant term in $g(n)$ is $7n$.

→ establish the inequality.

→ we want to find constants c and n_0 such that:

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

(3) Simplify the inequality:-

→ ignore the lower term 5 for larger

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

→ 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 $n \geq n$, the inequality holds:

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

we have shown that there exist constants (2) and $n_0 = n$ such that for all $n \geq n_0$.

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

Thus, we can conclude that :-

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

in Ω notation. The dominant term $2n^2$ in $f(n)$ clearly grows faster than $7n$. Hence

$$f(n) = \Omega(n^2).$$

However, for the specific comparision asked $f(n) = \Omega(7)$ is also correct.

Showing that $f(n)$ grows atleast as fast as $7n$.

Scanned with OKEN Scanner