Assignment - 9

NAME: R. Varun Kumar
REG: 192372040
Sub Code: CSA0670

1) Solve the following recurrence relations.

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

Solution:

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

$\Rightarrow x(1) = 0$

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

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

$\Rightarrow x(n) = x(3) + 5 = 10 + 5 = 15$

2) identify the pattern (or) The general term

$\Rightarrow$ The first term $x(1) = 0$

The common difference $d = 5$

The general formula for the n-th term of an AP is

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

Substituting the given values

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

$\therefore$ The solution is

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

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

Solution:

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

$\Rightarrow x(1) = 4$

$\Rightarrow x(2) = 3x(1) = 3 \cdot 4 = 12$

$\Rightarrow x(3) = 3x(2) = 3 \cdot 12 = 36$

$\Rightarrow x(4) = 3x(3) = 3 \cdot 36 = 108$

2) identify the general term:

$\Rightarrow$ The first term is $x(1) = 4$

$\Rightarrow$ The common ratio $r = 3$

The general formula for the n-th term of a GP is

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

substituting the given values

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

$\therefore$ 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 $n = 2^k$)

Solution:

for $n = 2^k$, we can write recurrence in terms of k.

1) substitude $n = 2^k$ 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:

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

lets sum the series:

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

since $x(1) = 1$:

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

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

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

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

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

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

∴ 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 terms 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

$\Rightarrow x(1) = 1$

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

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

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

3) identify the general term:

we observe that:

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

summing up the series.

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

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

∴ the solution is:

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

Evalute the following recurrences complexity.

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

Solution

The recurrence relation can be solved using iteration method

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

2) iterate the recurrence:

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

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

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

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

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

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

$\therefore$ 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.

Solution:

This recurrence can be solved using the master theorem for devide-and-conquer recurrence of the form:

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

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

The master lets determine the value of $\log_b a$:

$$\log_b a = \log_3 2$$

using the properties of logarithms

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

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

$f(n) = O(n)$

$n = n^1$

since $1 > \log_3 2$, we are in the third case of the master theorem:

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

$\therefore$ The solution is:

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

3) Consider the following recurrence algorithm?

```
min1(A[0.... n-1])
if n=1 return A[0]
Else temp=min1(A[0.... n-2])
     if temp<=A(n-1) return temp
Else
     Return A[n-1]
```

a) what does this algorithm compute?

Solution:

The given algorithm, 'Min1(A[0.... n-1])' computes the minimum value in the array 'A' from index '0' to 'n-1'. it does this by recursively finding the minimum value in the sub array 'A[0... 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's basic operation count and solve it.

Solution:

To determine the recurrence relation for the algorithm's basic operation count, let's analyse the steps involved in the algorithm. The basic operations are the comparisons and function calls.

Recurrence relation setup

1) Base case: when $n=1$, the algorithm performs a single operation to return $A[0]$

2) Recursive case: when $n>1$, the algorithm:

⇒ makes a recursive call to $\min(A[0 \dots n-2])$;

⇒ performs a comparison between temp and $A[n-1]$

Let $T(n)$ represent the no. of basic operation the algorithm performs for an array of size n.

1) Base case:

$$T(1)=1$$

2) Recursive case

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

Here $T(n-1)$ accounts for the operations performed by the recursive call to $\min(A[0 \dots n-2])$; and the $+1$ accounts for the comparison between temp and $A[n-1]$

To solve this recurrence relation, we can use iteration method.

$$\begin{aligned}
T(n) &= T(n-1)+1 \\
&= (T(n-2)+1)+1 \\
&= ((T(n-3)+1)+1)+1 \\
&= \dots \\
&= T(1)+(n-1) \\
&= 1+(n-1) \\
&= n
\end{aligned}$$

∴ The solution is

$$T(n) = n$$

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

n) Analyze the order of growth.

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

Solution:

To analyze the order of growth and use the Ω notation, we need to compare the given functions $F(n)$ 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 of a function. specifically, $F(n) = \Omega(g(n))$ means that for sufficiently large n, $F(n)$, grows at least as fast as $g(n)$

formally, $F(n) = \Omega(g(n))$ if there exist positive constants c and n_0 such that for all $n \geq n_0$:

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

Lets 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 it grows faster than the constant term 5 as n increases.

⇒ The dominant term in $g(n)$ is $7n$

2) establish the inequality:

Want to find constants c and n_0 such that:

$2n^2+5 \geq c \cdot 7n$ for all $n \geq n_0$ (3)

3) Simplify the inequality:

→ ignore the lower order terms for larger

$2n^2 \geq 7cn$

⇒ Divide both sides by n.

$2n \geq 7c$

⇒ Solve for n:

$n \geq \frac{7c}{2}$

4) Choose constants

Let $c=1$

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

∴ for $n \geq 4$, the inequality holds:

$2n^2+5 \geq 7n$ for all $n \geq 4$

We have shown that there exist constants $c=1$ and $n_0=4$ 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 comparison asked $f(n) = \Omega(7n)$ is also correct

showing that $f(n)$ grows at least as fast as $7n$.