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 n^{th} term of an A.P 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

$$x(1) = 4$$
$$x(2) = 3x(1) = 3\cdot 4 = 12$$
$$x(3) = 3x(2) = 36$$
$$x(4) = 3x(3) = 108$$

2. Identify the general term

→ The first term $x(1) = 4$

→ 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

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

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 1st 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

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

since $x(1) = 1$!

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

The geometric series with the term $a = 2$ and the Last term 2^K except for the additional 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 $n = K$

2 Evaluate the following recurrences complexity.

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

The recurrence relation can be solved using iteration method.

(i) Substitute $n=2^K$ in the recurrence

(ii) Iterate the recurrence

$$\text{for } K=0 : T(2^0) = T(1) = T(1)$$

$$K=1 : T(2^1) = T(1)+1.$$

$$K=2 : T(2^2) = T(8) = T(n)+1 \{T(1)+1\}+1 = T(1)+2$$

$$K=3 : T(2^3) = T(8) = T(n)+1 \{T(1)+2\}+1 = T(1)+3$$

3. generalize the pattern

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

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

$$T(n) = T(2K) = T(1)+\log_2 n$$

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

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

The solution is $T(n) = O(\log n)$

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

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

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

$$a=2, b=3 \quad \text{and } f(n)=cn$$

lets determine the value of $\log_b a$

$$\log_b a = \log_3 2$$

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

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

$$x(1) = 1$$

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

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

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

3. Identify the general term

we observe that

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

Summing, up the series

$$x(3^{K}) = 1+1+1+ \cdots + 1$$

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

The Solution is $x(3^{K}) = K+1$

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, $\log_3 2$ we are in the third case of masters thoerem

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

The solution is:

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

3\. Consider the following recurrence algorithm?

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

if $n = 1$ return A (0)

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.

The given algorithm, min[A[0, ... n-1]] computes the minimum value in the array 'A' from index '0' for 'n-1' if does this by recursively finding the minimum value in the subarray A [0 ... n-2] and then comparing it with the last element 'A [n-1] to determine the overall maximum value.

b. Set up a recurrence relation for the algorithm basic operation count and solve it

To determine the recurrence relation for the algorithms basic operation count, let's analyze the steps invoved in the algorithm the basic operation are the comparison 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 $\min(A[0 \cdots n-2])$; performs a comparison b/w temp and $A[n-1]$

1. Base case:

$$T(1) = 1$$

2. Recursive case:-

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

Here $T(n-1)$ accounts for the operations perfomed by the recursive call to $\min(A[0 \cdots n-2])$; and the $+1$ accounts for the comparision b/w temp and $A[n-1]$

To solve this recurrence relation we can use iteration method:

$$T(n) = T(n-1)+1$$
$$= (T(n-2)+1)+1$$
$$= (T(n-3)+1)+1)+1$$
$$= 1+(n-1)$$
$$= n$$

The Solution is

$$T(n) = n.$$

This means the algorithm performs n basic operations 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 $\Omega g(n)$ 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), g(n) w

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

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 then the constant terms as n increases
   → the dominant term in $g(n)$ is $7n$
2. 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 order terms for larger

$2n^2 \geq 7cn$

→ Divide both sides by n

$2n \geq 7c$

→ Solve from 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

$2n^2+6 \geq 7n$ for all $n \geq n$

we use shown the there exist constants $c \geq 1$ and $n_0 = n$

Such that for all $n \geq n_0$

$2n^2+6 \geq 7n$

Thus, we can conclude that

$f(n) = 2n^2+6 = \Omega(7n)$

in Ω notation the domain term $2n^2 f(n)$ clearly grows faster than n, 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 7n