1. **Find the value of T(2) for the recurrence relation T(n) = 3T(n-1) + 12n, given that T(0)=5.**

To find T(2), we can directly substitute n = 2 into the recurrence relation:

T(2) = 3T(2-1) + 12(2) T(2) = 3T(1) + 24

Now, we need to find T(1). Using the recurrence relation again:

T(1) = 3T(1-1) + 12(1) T(1) = 3T(0) + 12

Since T(0) = 5, we can substitute:

T(1) = 3(5) + 12 T(1) = 27

Now, we can substitute T(1) back into the equation for T(2):

T(2) = 3(27) + 24 T(2) = 81 + 24 T(2) = 105

Therefore, **T(2) = 105**.

2. Solving Recurrence Relations Using Substitution Method

a. T(n) = T(n-1) + c

Let's substitute T(n-1) with its definition:

T(n) = T(n-2) + c + c

Continuing this process:

T(n) = T(n-3) + c + c + c

...

T(n) = T(0) + cn

Substituting T(0) = 5:

T(n) = 5 + cn

Therefore, T(2) = 5 + 2c

b. T(n) = 2T(n/2) + n

Substituting T(n/2):

T(n) = 2(2T(n/4) + n/2) + n

Simplifying:

T(n) = 4T(n/4) + 2n

Continuing this process:

T(n) = 8T(n/8) + 3n ... T(n) = 2^kT(n/2^k) + kn

When $n = 2^k$, we get:

$T(2^k) = 2^kT(1) + k2^k$

Assuming $T(1)$ is a constant:

$T(2^k) = O(k2^k)$

Therefore, $T(n) = O(n\log n)$

c. $T(n) = 2T(n/2) + c$

Similar to case b, we can derive:

$T(n) = 2^kT(n/2^k) + ck$

When $n = 2^k$, we get:

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

Assuming $T(1)$ is a constant:

$T(2^k) = O(2^k)$

Therefore, $T(n) = O(n)$

d. $T(n) = T(n/2) + c$

Substituting $T(n/2)$:

$T(n) = T(n/4) + c + c$

Continuing this process:

$T(n) = T(n/2^k) + kc$

When $n = 2^k$, we get:

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

Assuming $T(1)$ is a constant:

$T(2^k) = O(k)$

Therefore, $T(n) = O(\log n)$

3. Solving Recurrence Relations Using the Recursive Tree Approach

a. $T(n) = 2T(n-1) + 1$

The recursive tree will have a depth of n, with each level having 2^i nodes. The work done at each level is 1. Therefore, the total work is:

$T(n) = \Sigma(i=0 \text{ to } n-1)\ 2^i$

Using the geometric series formula:

$T(n) = 2^n - 1$

Therefore, $T(2) = 2^2 - 1 = 3$

b. $T(n) = 2T(n/2) + n$

The recursive tree will have a depth of logn, with each level having 2^i nodes. The work done at each level is in. Therefore, the total work is:

$T(n) = \Sigma(i=0 \text{ to } \log n-1)\ 2^i * (n/2^i)$

Simplifying:

$T(n) = n * \Sigma(i=0 \text{ to } \log n-1)\ 1$

$T(n) = n * \log n$

Therefore, $T(2) = 2 * \log 2 = 2$