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

Soln

$$T(n) = 3T(n-1) + 12n \quad \text{--- (1)}$$

To find $T(2)$, $n=2$ can be put in (1)

$$T(2) = 3T(2-1) + 12 \times 2$$

$$\Rightarrow T(2) = 3T(1) + 24$$

for $T(1)$ calculation, $n = 1$ in (1)

$$T(1) = 3T(1-1) + 12 \times 1$$

$$\Rightarrow T(1) = 3T(0) + 12$$

Since $T(0) = 5$

$$T(1) = 3 \times 15 + 12 = 57$$

$$\Rightarrow T(2) = 3 \times 57 + 24$$

$$T(2) = 171 + 24$$

$$T(2) = 195$$ Ans

2023 11 26 19:41

② Given a recurrence relation, solve it using substitution method:

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

Since base condition is not mentioned, we can assume that

$$T(n) = \begin{cases} 1 & n = 1 \\ T(n-1) + C & n > 1 \end{cases}$$

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

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

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

$\vdots$

$T(1) = T(0) + C$

$T(n) = T(n-2) + C + C = T(n-2) + 2C$

$T(n) = T(n-3) + C + C + C = T(n-3) + 3C$

$T(n) = T(n-4) + C + C + C + C = T(n-4) + 4C$

$\Rightarrow T(n) = T(n-K) + KC$

Recursion will stop when $n - K = 1 \Rightarrow K = n-1$

$\Rightarrow T(n) = T(n-n+1) + (n-1)C$

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

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

$\Rightarrow$ Time complexity is $O(n)$ as $O(nC) \approx O(n)$

Ans

2023 11 26 21:03

b) $T(n) = 2^1 T(n/2) + 1 \cdot n$

$T(n/2) = 2T(n/4) + \frac{n}{2} = 2T\left(\frac{n}{2^2}\right) + \frac{n}{2}$

$T(n) = 2\left\{2T\left(\frac{n}{2^2}\right) + \frac{n}{2}\right\} + n = 2^2 T\left(\frac{n}{2^2}\right) + 2n$

$T(n/4) = 2T(n/8) + \frac{n}{4} = 2T\left(\frac{n}{2^3}\right) + \frac{n}{2^2}$

$T(n) = 2^2\left\{2T\left(\frac{n}{2^3}\right) + \frac{n}{2^2}\right\} + 2n$

$T(n) = 2^3 T\left(\frac{n}{2^3}\right) + n + 2n = 2^3 T\left(\frac{n}{2^3}\right) + 3n$

$\cancel{= 4T\left(\frac{n}{2^3}\right) + \frac{5n}{2}}$

$\Rightarrow$ for k^{th} term

$2^k T\left(\frac{n}{2^k}\right) + kn$

The ~~iter~~ recursion will stop ~~when~~ at base condition

$\Rightarrow 2^k T\left(\frac{n}{2^k}\right) + kn \Rightarrow \frac{n}{2^k} = 1$

$\Rightarrow n = 2^k \Rightarrow 2^k = n \Rightarrow k \log_2 2 = \log_2 n$

$\Rightarrow k \times 1 = \log_2 n \Rightarrow k = \log_2 n$

$\Rightarrow$ Time complexity is $O(n \log_2 n)$ Ans

As $T(n) = 2^{\log_2 n} \times T(1) + n \log_2 n$

$T(n) = n \times 1 + n \log_2 n = n + n \log_2 n$

2023 11 26 20:17

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

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

$\Rightarrow T(n) = 2 \cdot 2 \cdot T(n/2^2) + c + c = 2^2 \cdot T(n/2^2) + 2c$

$\Rightarrow$ k^{th} term will be

$2^k T\left(\frac{n}{2^k}\right) + kc$

Now, base condition will be achieved. If

$n/2^k = 1 \Rightarrow n = 2^k \Rightarrow 2^k = n$

$\Rightarrow k \log_2 2 = \log_2 n \Rightarrow \boxed{k = \log_2 n}$

$\Rightarrow T(n) = 2^{\log_2 n} + T\left(\frac{n}{2^{\log_2 n}}\right) + (\log_2 n) \cdot c$

$T(n) = n + T(1) + c \log_2 n$

$T(n) = n + 1 + c \log_2 n$

$\Rightarrow$ Time complexity will be $O(n)$ by considering the worst case scenario

$n > c \log_2 n$

Ans

d) $T(n) = T(n/2) + C$

$T(n/2) = T(n/2^2) + C$

$T(n) = T(n/2^2) + 2C$

$\Rightarrow$ for K-term

$T(n/2^k) + KC$

The base condition will be achieved If $n/2^k = 1$

$\frac{n}{2^k} = 1 \Rightarrow 2^K = n \Rightarrow K = \log_2 n$

$\Rightarrow T(n) = T(1) + c\log_2 n$

$\Rightarrow$ Time complexity is $O(\log_2 n)$

Ans