Recurrence Relations (Assignment 4)

1) Find the value of T(2) for the recurrence relation

$T(n) = 3T(n-1) + 12n$, given that $T(0) = 5$

Ans. $T(1) = 3T(n-1) + 12n$

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

$\Rightarrow 3T(0) + 12$

$\Rightarrow 3 \times 5 + 12$

$\Rightarrow 15 + 12$

$\Rightarrow 27$

So, the value of $T(2) = 27$

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

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

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

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

$T(n) = T(n-1) + T(n-2) + T(n-3) + \cdots$
$\quad T(n-k-1) + T(n-k) + T(n-k+1)$
$\quad + T(n-k+2)$

(c) $T(n) = 2T(n/2) + c$; $T(n) = 1$, when $n = 1$

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

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

$T(n/2^2) = \boxed{2T(n/2^3) + c + 2c}$

$= 2T(n/2^3) + 3c$

let it continues for k time

T~~(n)~~ =

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

so, now $n/2^k = 1$

$n = 2^k$

$\log_2 n = \log_2 2^k$

$\log_2 n = k$

$T(n) = 2T(n/2^{\log_2 n}) + (\log_2 n) \times c$

$\Rightarrow 2T(n/n) + \log_2 n\, c$

$\Rightarrow 2T(1) + \log_2 n\, c$

$\Rightarrow 2 + \log_2 n\, c$

Time complexity $= O(\log n)$

(d)

$\Rightarrow c\,(1+2+3+4+\cdots+(n-1)+n)$

$\Rightarrow c\,\frac{n(n+1)}{2}$

$\Rightarrow c\left(\frac{n^2+n}{2}\right)$

Time complexity $= O(n^2)$

(b) $T(n) = 2T(n/2) + n$; $T(n)=1,\ n=1$

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

$$\boxed{T(n) = 2T(n/2^2) + n/2 + n}$$

$T(n/2^2) = 2T(n/2^3) + n/2^2 + n/2 + n$

$\downarrow$ k times so, $n/2^k = 1$

$n = 2^k$

$\log_2 n = \log_2 2^k$

$k = \log_2 n$

$T(n) = 2T(n/2^k) + n/2^{k-1} + n/2^{k-2} + \cdots + n/2 + n$

$\Rightarrow 2[T(n/2^{\log_2 n}) + \frac{n}{2^{(\log_2 n-1)}} + \frac{n}{2^{(\log_2 n-2)}} + n/2 + n]$

$\Rightarrow 2T(1) + (1/2)^0 + (1/2)^1 + (1/2)^2 + \cdots (1/2)^{\log_2(n)}$

$\Rightarrow$ we know $T(1) = 1$

Here $a = 1$ $\quad r < 1$
$r = 1/2$

$$S_n = \left[\frac{a(1-r^n)}{1-r}\right] n$$

$$\Rightarrow \left[\frac{1\left(1-(1/2)^{\log_2 n}\right)}{1-1/2}\right] \times n$$

$\Rightarrow$ n·[constant]

Time complexity = $O(n)$

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

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

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

$\downarrow$ k times

so $n - k = 1$
$\Rightarrow n - 1 = k$

$T(n) = T(n-k) + kc$
$= T(n-n+1) + (n-1)c$
$\Rightarrow T(1) + (n-1)c$
$\Rightarrow 1 + (n-1)c$

Time complexity = $O(n)$

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

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

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

$$T(n/2^2) = T(n/2^3) + 3c$$

If we repeat this k time

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

$$n/2^k = 1$$

$$n = 2^k$$

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

$$\log_2 n = k$$

$$T(n) = T(n/2^{\log_2 n}) + \log_2 n \times c$$

$$\Rightarrow T(n/n) + \log_2 n \times c$$

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

$$\Rightarrow 1 + \log_2 n \times c$$

Time complexity $= O(\log n)$

3) Given a recurrence relation, solve it using the recursive tree approach:

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

Recursive Tree

$$n \longrightarrow 1$$

$$(n-1)$$

$$(n-1-1) = (n-2) \longrightarrow 2$$

$$(n-2-1) = (n-3) \longrightarrow 3$$

let it continues upto k times

So $n-k = 1$

$n-1 = k$

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

$$\Rightarrow 2T(n-(n-1)) + (n-1)$$

$$\Rightarrow 2T(1) + (n-1)$$

$$\Rightarrow 2 + (n-1)$$

Time complexity $= O(n)$

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

Recursive Tree

$n \rightarrow n = n$

$n/2 \rightarrow n + n/2 = 3n/2$

$n/2^2 \rightarrow n + n/2 + n/2^2 = 4n + 2n + n$
$= 7n/2^2$

$n/2^3 \rightarrow n + n/2 + n/2^2 + n/2^3$

$= n + n/2 + n/4 + n/8$

$= \frac{8n + 4n + 2n + n}{8}$

$= 15n/8$

Let it continues for k times

so, $n/2^k = 1$

$\Rightarrow n = 2^k$

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

$\Rightarrow \log_2 n = k$

$T(n) = \cancel{\ldots}$

$T(n) = n\left(\left(\tfrac{1}{2}\right)^0 + \left(\tfrac{1}{2}\right)^1 + \left(\tfrac{1}{2}\right)^2 + \cdots + \left(\tfrac{1}{2}\right)^k\right)$

$\Rightarrow n\left(1 + \tfrac{1}{2} + \tfrac{1}{4} + \cdots + \left(\tfrac{1}{2}\right)^{\log_2 n}\right)$

Here $a = 1$
$r = 1/2$ $\quad r < 1$

$S = \frac{a(1 - r^k)}{1 - r} \times n$

$$O\left(1 \cdot \frac{1\left(1-\left(\frac{1}{2}\right)^{\log_2 n}\right)}{1-\frac{1}{2}}\right)$$

$\Rightarrow O(1) \times n$

$\Rightarrow$ Time complexity $= O(n)$