13 January 2025 12:05

Recursion - A f^n calling itself.

$A(n) \rightarrow T(n)$.

$A(n/2)$ $T(n/2)$

$T(n)$

A (int n)
{
if(n>1) return A(n/2) + A(n/2);
→ return 1; → 1
}

1, $T(n/2)$, $T(n/2)$

$$T(n) = \begin{cases} 1 + 2T(n/2), & n>1 \\ 1, & n=1 \end{cases}$$

→ Recurrence Relation.

A (int n) {
if(n>1) return A(n-1);
return 1; → 1

Let, $A \longrightarrow T(n)$

$1 + T(n-1)$

$A(n-1) \rightarrow T(n-1)$

$$T(n) = \begin{cases} 1 + T(n-1), & n > 1 \\ 1, & n = 1 \end{cases}$$ Recurrence Relatⁿ.

Back Substitution Method –

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

$T(n) = 1 + T(n-1)$ —

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

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

$\vdots$

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

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

$T(1) = 1$

$T(n) + T(\cancel{n-1}) + T(\cancel{n-2}) + \cdots + T(\cancel{3}) + T(\cancel{2}) + T(\cancel{1}) =$

$1 + T(\cancel{n-1}) + 1 + T(\cancel{n-2}) + 1 + T(\cancel{n-3}) + \cdots + 1 + T(\cancel{2}) + 1 + T(\cancel{1}) + 1$

$\Rightarrow T(n) = n \times 1 = n = O(n).$

Q

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

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

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

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

$\vdots$

$T(3) = 3 + T(2)$

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

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

$T(1) = 1$

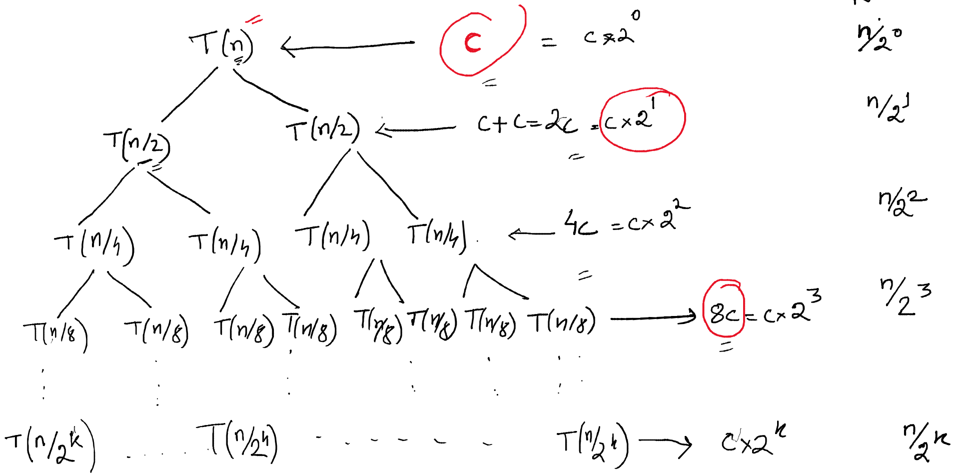
$$T(n) + \cancel{T(n-1)} + \cancel{T(n-2)} + \cdots + \cancel{T(3)} + \cancel{T(2)} + \cancel{T(1)} = n + \cancel{T(n-1)} + (n-1) + \cancel{T(n-2)} + (n-2) + \cancel{T(n-3)} + \cdots + 3 + \cancel{T(2)} + 2 + \cancel{T(1)} + 1 =$$

$$\Rightarrow T(n) = 1 + 2 + 3 + \cdots + (n-2) + (n-1) + n.$$

$$= \frac{n \times (n+1)}{2} = O(n^2) //$$

Recursion tree method

$$T(n) = \begin{cases} 2T(n/2) + c, & n > 1 \\ c, & n = 1 \end{cases}$$



Recursion stop → $\frac{n}{2^k} = 1$

$\Rightarrow n = 2^k$

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

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

$\Rightarrow k = \log_2 n$

GP sum $= \frac{a(r^n - 1)}{r-1}$

Total cost

$c + 2c + 4c + 8c + \ldots\ldots + 2^k \times c$

$= c2^0 + c2^1 + c2^2 + c2^3 + \ldots \; c2^k$

$= c(2^0 + 2^1 + \ldots + 2^k)$

$= c \times \frac{1 \times (2^{k+1} - 1)}{2-1}$

$= c \times 2^k \cdot 2 - c$

$= c \cdot 2^k - c$

$\approx 2^k$

$= 2^{\log_2 n}$

$= n^{\log_2 2}$

$= n$

$\Rightarrow TC = O(n)$.