Recurrence Relation:

When an algorithm contains a recursive call to itself then its running time can be described by mathematical relation called recurrence relation or recurrence equation. Recurrence relations are used to determine the running time of a recursive program.

1. Substitution method
2. Master theorem

Example 1:

$void\ \ display(\text{int}\ \ n)------T(n)$

$\{$

$if\,(n>0)$

$\{$

$pr\,\text{int}\ \ f\,("\%d",n);------(1)$

$display\,(n-1);---------T(n-1)$

$\}$

$\}$

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

So recurrence relation is:

$$T(n) = \begin{cases} 1 \quad if \;\; n = 0 \\ T(n-1)+1 \qquad n > 0 \end{cases}$$

Back Substitution Method:

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

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

$$substitute \;\; (2) \;\; in(1), we \;\; get$$

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

$$Similarly, T(n) = T(n-3) + 3$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$Continuing \; k \;\; times$$

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

$$now \;\; put \;\; n - k = 0 \Rightarrow k = n$$

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

$$\Rightarrow T(n) = O(n)$$

Example 2:

$$void\ \ loop\_test(\text{int}\ \ n)------T(n)$$

$$\{$$

$$if\,(n>0)-------------(1)$$

$$\{$$

$$for(i=0;i<n;i++)------(n+1)$$

$$\{pr\,\text{int}\ f\,(\text{"}\%d\text{"},i);\}------(n)$$

$$\}$$

$$loop\_test(n-1);---------T(n-1)$$

$$\}$$

$$\begin{aligned}\Rightarrow T(n) &= T(n-1)+2n+2\\ &= T(n-1)+n\end{aligned}$$

Recurrence relation is:

$$T(n)=\begin{cases}1 & n=0\\ T(n-1)+n & for\ \ n>0\end{cases}$$

Substitution Method:

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

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

$$substitute\ \ (2)\ in(1), we\ \ get$$

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

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

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$Continuing\ k\ \ times$$

$$T(n) = T(n-k) + (n-(k-1)) + (n-(k-2)) + \ldots(n-1) + n$$

$$now\ \ put\ \ n-k = 0 \Rightarrow k = n$$

$$T(n) = T(n-n) + 1 + 2 + \ldots + (n-1) + n = T(0) + \frac{n(n+1)}{2}$$

$$\Rightarrow T(n) = \theta(n^2)$$

Example 3:

$void\ \ loop\_product(\text{int}\ \ n)------T(n)$

$\{$

$if\,(n>0)-------------(1)$

$\{$

$\quad for(i=1;i<=n;i=i*2)-----\log(n)+1$

$\quad \{print\,f\,("\%d",i);\}------(\log n)$

$\}$

$loop\_product(n-1);---------T(n-1)$

$\}$

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

Recurrence relation is:

$$T(n)=\begin{cases}1 & n=0\\ T(n-1)+\log n & for\ \ n>0\end{cases}$$

Substitution Method:

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

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

$$substitute\ \ (2)\ in(1), we\ \ get$$

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

$$Similarly, T(n) = T(n-3) + \log(n-2) + \log(n-1) + \log n$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$Continuing\ k\ \ times$$

$$T(n) = T(n-k) + \log(n-(k-1)) + \ldots + \log(n-1) + \log n$$

$$now\ \ put\ \ n-k = 0 \Rightarrow k = n$$

$$T(n) = T(n-n) + \log 1 + \log 2 + \ldots + \log(n-1) + \log n$$

$$= T(0) + \log(1.2.3\ldots n) = \log(n!)$$

$$\Rightarrow T(n) = \theta(n \log n)$$

$$Since\ UB\ \ for\ \ n! = n^n \Rightarrow UB\ \ for\ \log(n!) = \log(n^n) = n \log n$$

How to simplify the recurrence relation of a decreasing function:

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

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

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

$$T(n) = T(n-1) + \log n --------- O(n \log n)$$

$$T(n) = T(n-2) + 1 \rightarrow \frac{n}{2} \rightarrow O(n)$$

$$T(n) = T(n-50) + n --------- O(n^2)$$