$\underline{Handshake\ Problem - Time\ Complexity}$

$\underline{Using\ substitution\ method}$

$As\ we\ see\ the\ program\ , the\ return\ value\ is: return\ handshake(n-1)+(n-1)\ , there\ is\ two\ approaches:$

$\underline{First\ Approach}:$

$Take\ n = 4$

$Hence\ Base\ Cases:$

$n = 0\ , has\ 0\ handshake\ and\ it\ returns\ 0\ takes\ 1\ unit\ of\ time.$

$n = 1\ , has\ 0\ handshake\ and\ it\ returns\ 0\ takes\ 1\ unit\ of\ time.$

$When\ n > 1, here\ 4, therefore\ total\ push\ inside\ stack:$

$n = 1$	$Value\ of\ n.$
0	$Return\ Value.$
$n = 2$	$Value\ of\ n.$
$+(n-1\ i.e. 2-1=1)$	$Return\ Value$
$n = 3$	$Value\ of\ n$
$+(n-1\ i.e. 3-1=2)$	$Return\ Value$
$n = 4$	$Value\ of\ n$
$+(n-1\ i.e. 4-1=3)$	$Return\ Value$
$Hand\ Shake(n=4)$	$Main\ Func()$

$Run\ 1\ unit\ of\ time\ (Base\ Case)$

$Runs\ T(n-1)\ i.e.\ T(4-1)\ times$

$Hence\ we\ get\ our\ relation$:

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

2nd Approach:

For base cases remain same , Lets take the stack representation of push:

$n = 1$	$Value\ of\ n.$
0	$Return\ Value.$
$n = 2$	$Value\ of\ n.$
$+(n - 1\ i.e. 2 - 1 = 1)$	$Return\ Value$
$n = 3$	$Value\ of\ n$
$+(n - 1\ i.e. 3 - 1 = 2)$	$Return\ Value$
$n = 4$	$Value\ of\ n$
$+(n - 1\ i.e. 4 - 1 = 3)$	$Return\ Value$
$Hand\ Shake(n = 4)$	$Main\ Func()$

Runs

$T(n - 1)$

$i.e.$

$T(4 - 1)$

times

Lets take the addition operation in pop:

$n = 1$	***Value of n.***
0	***Return Value.***
$n = 2$	***Value of n.***
$0 + 1 = 1$	***Return Value***
$n = 3$	***Value of n***
$1 + 2 = 3$	***Return Value***
$n = 4$	***Value of n***
$3 + 3 = 6$	***Return Value***
Hand Shake$(n = 4)$	***Main Func***$()$

These addition takes constant time c i.e. ≈ 1 ***unit.***

$$T(n) = \underbrace{T(n-1)}_{\text{To Compute } H(n-1)} + \underbrace{1}_{\text{To add } H(n-1) \text{ by } n-1} \quad , \text{for } n > 1$$

Which gives:

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

$Moving\ with\ the\ 1st\ way$:

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

$And\ we\ know\ T(1) = 1\ , hence\ rewriting\ the\ linear\ recurrence\ equation\ as$:

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

$Or\ 2nd\ way$:

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

$$i.e. same\ T(n) = T(n-1) + 1$$

$$Therefore\ , continuing\ with\ the\ equation: T(n-1) + 1,$$

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

$$Substituting\ this\ in\ T(n)\ we\ get:$$

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

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

$$Substituting\ this\ in\ T(n)\ we\ get:$$

$$T(n) = \left((T(n-3) + 1)\ + 1\right)\ + 1$$

$Now\ if\ it\ runs\ upto\ `i`\ times\ we\ get$:

$Now\ ,T(n-i)=T(n-i-1)+1$

$Substituting\ this\ in\ T(n)\ we\ get$:

$$T(n)=T(n-i-1)+1+1+1+\cdots i\ times$$

$When\ i=n-1\ we\ get$:

$T(n)=T(n-(n-1)-1)+1+1+1+\cdots(n-1)times$

$=T(n-n+1-1)+1+1+1\ \ldots.(n-1)times$

$=T(0)+1+1+1\ \ldots.\ \ (n-1)times$

$And\ we\ know\ T(0)=1\ along\ with\ T(1) is\ 1\ and\ as\ 1\ is\ added\ to\ 1+1+1+\cdots(n-1)times\ \ it\ becomes$:

$=1+1+1+\cdots n\ times$

$i.e. 1\times n=n\ and\ time\ complexity\ is$:

$=O(n)$.

****