20.2. TIME COMPLEXITY CALCULATION

FOR LOOP (EG-1).

$//outer\ loop\ executed\ \ n\ times$

$for(i = 1; i \leq n; i++)\{$

$\quad //inner\ loop\ executes\ n\ times$

$\quad for(j = 1; j \leq i; j++)\{$

$\qquad c = c + 1\ ;\ //\ constant\ time.$

$\quad \}$

$\}$

SOLUTION:

$1. Inner\ most\ loop's\ statement \Longrightarrow c = c + 1\ which\ runs\ at$

$O(1)\ time\ i.e.\ 1\ unit\ of\ time\ \ .$

$2. No.\ of\ inputs\ in\ outer\ for\ loop\ takes\ 1\ to\ n\ times.$

$lets\ see\ the\ inner\ loop\ and\ runtime\ of$

$inner\ loop's\ statement.$

$$f(1) \le c \times n \Rightarrow when\ i = 1$$

$$f(1) \le c \times i \Rightarrow when\ j = 1$$

$$c = c + 1\ runs\ 1\ unit\ of\ time.$$

$[Hence\ ,total\ amount\ of\ taken\ to\ run\ (c = c + 1\ )$
$is\ 1\ unit\ of\ time]$

$$f(2) \le c \times n \Rightarrow when\ i = 2$$

$$f(1) \le c \times i \Rightarrow when\ j = 1$$

$$c = c + 1\ runs\ 1\ unit\ of\ time.$$

$$f(2) \le c \times i \Rightarrow when\ j = 2$$

$$c = c + 1\ runs\ 1\ unit\ of\ time.$$

$[Hence\ ,total\ amount\ of\ taken\ to\ run\ \ (c = c + 1\ )$
$is\ (1 + 1 = 2\ )unit\ of\ time]$

$$f(3) \le c \times n \Rightarrow when\ i = 3$$

$$f(1) \le c \times i \Rightarrow when\ j = 1$$

$$c = c + 1\ runs\ 1\ unit\ of\ time.$$

$$f(2) \le c \times i \Rightarrow when\ j = 2$$

$$c = c + 1\ runs\ 1\ unit\ of\ time.$$

$$f(3) \le c \times i \Rightarrow when\ j = 2$$

$$\boldsymbol{c = c + 1\ runs\ 1\ unit\ of\ time.}$$

$[\boldsymbol{Hence\ , total\ amount\ of\ taken\ to\ run\ \ (c = c + 1\ )}$
$\boldsymbol{is\ (1 + 1 + 1 = 3\ ) unit\ of\ time}]$

…………

$\boldsymbol{f(n) \leq c \times n \Longrightarrow when\ i = n}$

$\boldsymbol{f(1) \leq c \times i \Longrightarrow when\ j = 1}$

$\boldsymbol{c = c + 1\ runs\ 1\ unit\ of\ time.}$

$\boldsymbol{f(2) \leq c \times i \Longrightarrow when\ j = 2}$

$\boldsymbol{c = c + 1\ runs\ 1\ unit\ of\ time.}$

$\boldsymbol{f(3) \leq c \times i \Longrightarrow when\ j = 2}$

$\boldsymbol{c = c + 1\ runs\ 1\ unit\ of\ time.}$

……

$\boldsymbol{f(n) \leq c \times i \Longrightarrow when\ j = n}$

$\boldsymbol{c = c + 1\ runs\ 1\ unit\ of\ time.}$

$[\boldsymbol{Hence\ , total\ amount\ of\ taken\ to\ run\ \ (c = c + 1\ )}$
$\boldsymbol{is\ (1 + 1 + 1 + \cdots n\ times = n\ ) unit\ of\ time}]$

$We\ have\ to\ see\ the\ number\ of\ times\ to\ calculate$

$time\ complexity.$

$1 + 2 + 3 + 4 + \cdots + n - 1 + n\ times$

$By\ arithmetic\ series(Arithmetic\ Progression$

$to\ find\ general\ term):$

$$\Rightarrow S(n) = \frac{n}{2}\big((2 \times a) + (\,(n-1) \times (d)\,\big)$$

$Where\ , a = First\ Term.$

$d = (\,T_n - T_{n-1})\ or\ it\ can\ be\ 2^{nd}\ \ term -$
$(minus)1^{st}\ term.$
$i.e. the\ common\ difference.$

$T_{n-1} = Second\ Last\ term\ \ \Rightarrow n - 1.$
$T_n = Last\ Term \Rightarrow n.$
$n - 1 = Second\ last\ term\ i.e. T_{n-1}.$

$Here\ d = T_n - T_{n-1} = n - (n-1) = 1$

$$\Rightarrow S(n) = \frac{n}{2}\big((2 \times 1) + (\,(n-1) \times (1)\,\big)$$

$$\Rightarrow S(n) = \frac{n}{2}(2 + n - 1\ \ )$$

$$\Rightarrow S(n) = \frac{n}{2}(1 + n\ \ )$$

$$= \frac{n(n+1)}{2} = \frac{n^2+n}{2} = \mathrm{O}\left(\frac{1}{2} \times n^2 + \frac{1}{2} \times n\right)$$

By Constant Rule: $\mathrm{O}(k \times n) = \mathrm{O}(n)$, *where k is constant*.

$$\mathrm{O}\left(\frac{1}{2} \times n^2 + \frac{1}{2} \times n\right) = \mathrm{O}\left(\frac{1}{2}(n^2+n)\right) = O(n^2)$$

Therefore time complexity of the program is :

$$= O(n^2)$$
