20.3. TIME COMPLEXITY CALCULATION FOR LOOP (EG-2).

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

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

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

$\quad for(j = 1; j \leq i/2; j++)\{$

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

$\}$

SOLUTION:

$1. Inner\ most\ loop's\ statement \Longrightarrow c = c + 1\ which\ runs\ at$
$O(1)\ time\ i.e.1\ unit\ of\ time\ \ .$

$2. No.of\ iterations\ \ in\ outer\ for\ loop\ takes\ 1\ to\ n\ times.$
$lets\ see\ the\ inner\ loop\ and\ runtime\ of$
$inner\ loop's\ statement.$

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

$$f(1) \leq c \times \tfrac{1}{2} \times i \Rightarrow when\ j = 1$$

$$c = c + 1\ executes\ in\ \frac{1}{2} \times 1\ unit\ of\ time\ .$$

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

$is\ 1\ unit\ of\ time]$

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

$$f(1) \leq c \times \tfrac{1}{2} \times i \Rightarrow when\ j = 1$$

$$c = c + 1\ executes\ in\ \frac{1}{2} \times 1\ unit\ of\ time\ .$$

$$f(2) \leq c \times \frac{1}{2} \times i \Rightarrow when\ j = 2$$

$$c = c + 1\ executes\ in\ \tfrac{1}{2} \times 1\ unit\ of\ time$$

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

$is \frac{1}{2} + \frac{1}{2} = \frac{2}{2}\ unit\ of\ time]$

....

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

$$f(1) \leq c \times \tfrac{1}{2} \times i \Rightarrow when\ j = 1$$

$$c = c + 1\ executes\ in\ \frac{1}{2} \times 1\ unit\ of\ time\ .$$

$$f(2) \leq c \times \frac{1}{2} \times i \Rightarrow when\ j = 2$$

$$c = c + 1\ runs \tfrac{1}{2} \times 1\ \ unit\ of\ time.$$

....

$$f(n) \leq c \times \frac{1}{2} \times i \Rightarrow when\ j = n$$

$$c = c + 1\ runs \tfrac{1}{2} \times 1\ unit\ of\ time.$$

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

$is \frac{1}{2} + \frac{1}{2} + \cdots + \frac{1}{2}\ to\ n\ times = \frac{n}{2}\ unit\ of\ time]$

$No. of\ units\ of\ time\ taken\ to\ run\ the\ inner\ most\ statement$

$$\frac{1}{2} + \frac{2}{2} + \frac{3}{2} + \cdots + \frac{n}{2} = \frac{1}{2}(1 + 2 + 3 + \cdots + n) = \sum_{i=1}^{n} \frac{1}{2} \times i\ (Arithmetic\ Series)$$

VERY IMPORTANT

****

$By\ arithmetic\ series(Arithmetic\ Progression$

$to\ find\ \ Sum\ of\ first\ 'n'\ \ term):$

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

$Where\ , a = First\ Term.$

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

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

PROOF OF ABOVE EQUATION

$if\ `\ l\ `\ \ is\ last\ term\ i.e. l = a + (n-1)d\ \ and\ the\ equation\ actually\ is:$

$$Sn = a + (a + d) + (a + 2d) + (a + 3d) + \cdots + (l - 2d) + (l - d) + l$$

$Rewriting\ the\ series\ in\ reverse\ additive\ order:$

$$Sn = l + (l - d) + (l - 2d) + \cdots + (a + 2d) + (a + d) + a$$

$Adding\ columnwise\ we\ get$:

$$2Sn = (a + l) + (a + l) + (a + l) + \cdots n\ times = n(a + 1)$$

$$\therefore S_n = \frac{n}{2}[a + l]$$

$$\therefore S_n = \frac{n}{2}[a + a + (n - 1)d]$$

$$\therefore S_n = \frac{n}{2}[2a + (n - 1)d]$$

AND HOW WE ARE GETTING $T_n = a + (n - 1)d\ \ or\ l = a + (n - 1)d.$

ARITHMETIC PROGRESSION: $A\ sequence\ (finitite\ or\ infinite\ )is\ called\ an$ $arithmetic\ progession\ abbreviated\ as\ A.P\ iff\ the\ difference\ of\ any\ term$ $from\ its\ preceeding\ term\ is\ finite.$

Say: $we\ have\ 1 + 2 + 3 + \cdots n\ , then\ T_{n+1} - T_n = 1,$
$where\ T_{n+1}\ is\ preceeding\ term\ and\ T_n\ is\ term\ substracted\ i.e.2 - 1 = 1,$
$3 - 2 = 1 \ldots..etc.$

$if\ there\ is\ a\ sequence\ of\ a_1, a_2, a_3, \ldots, a_n\ then\ a_1 + a_2 + a_3 + \cdots + a_n\ is\ called\ an$ $Arithmetic\ Series.$

GENERAL TERM OF A.P:

$Let\ a\ be\ the\ first\ term\ and\ d\ be\ the\ common\ difference\ of\ Arithmetic$ $Progression(A.P).Let\ T_1, T_2, T_3, \ldots, T_{n-1}, T_n = 1st, 2nd\ , 3rd, \ldots, nth\ terms$ $respectively\ , then\ we\ have$:

$T_2 - T_1 = d$
$T_3 - T_2 = d$
$T_4 - T_3 = d$
.....
$T_n - T_{n-1} = d$

ADDING THESE $n - 1\ equations\ we\ get$:
$T_n - T_1 = (n - 1)d\ \Rightarrow T_n = T_1 + (n - 1)d\ , where\ T_1 = a, then$
$T_n = a\ + (n - 1)d\ .$

LAST TERM OF A.P
$if\ T_n = l\ (last\ term), then\ l = a + (n - 1)d\ .$

$\boldsymbol{Series}$: $\boldsymbol{if\ the\ terms\ are\ connected\ by\ +(signs) we\ get\ a\ series.}$
$\boldsymbol{T_1 + T_2 + \cdots + T_n\ \ is\ called\ a\ series.}$

****

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

$$\Rightarrow S(n) = \frac{1}{2}(1 + 2 + 3 + \cdots + (n-1) + n)$$

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

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

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

$$= \frac{n(n+1)}{4} = \frac{n^2 + n}{4} = O\left(\frac{1}{4} \times n^2 + \frac{1}{4} \times n\right)$$

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

$$O\left(\frac{1}{4} \times n^2 + \frac{1}{4} \times n\right) = O\left(\frac{1}{4}(n^2 + n)\right) = O(n^2)$$

$\boldsymbol{Therefore\ time\ complexity\ of\ the\ program\ is:}$

$$= O(n^2)$$

SOME OBSERVATION:

$c = c + 1$ *inner most statement will execute depending upon the*

upper bound of inner most loop j *i.e.* $\frac{i}{2}$:

i.e. when for $i = 1, j \le \frac{1}{2}, c = c + 1$ *will execute* $\frac{1}{2}$ *times.*

when for $i = 2, j \le \frac{2}{2}, c = c + 1$ *will execute* $\frac{2}{2}$ *times.*

when for $i = n, j \le \frac{n}{2}, c = c + 1$ *will execute* $\frac{n}{2}$ *times.*

and we can too add up upper bound $g(n) = \{\frac{1}{2}, \frac{2}{2}, \frac{3}{2}, \dots, \frac{n}{2}\}$ *of*

inner most loop $\rightarrow j$ *as outer loop* $\rightarrow$ i *increment at each iteration*

i.e. $\frac{1}{2} + \frac{2}{2} + \frac{3}{2} + \cdots + \frac{i}{2}$ *since*

we are looking for upper bound and i will execute till `n` time

hence at $i = n$ *, we have* : $\frac{1}{2} + \frac{2}{2} + \frac{3}{2} + \cdots + \frac{n}{2}$. *This is also correct.*
