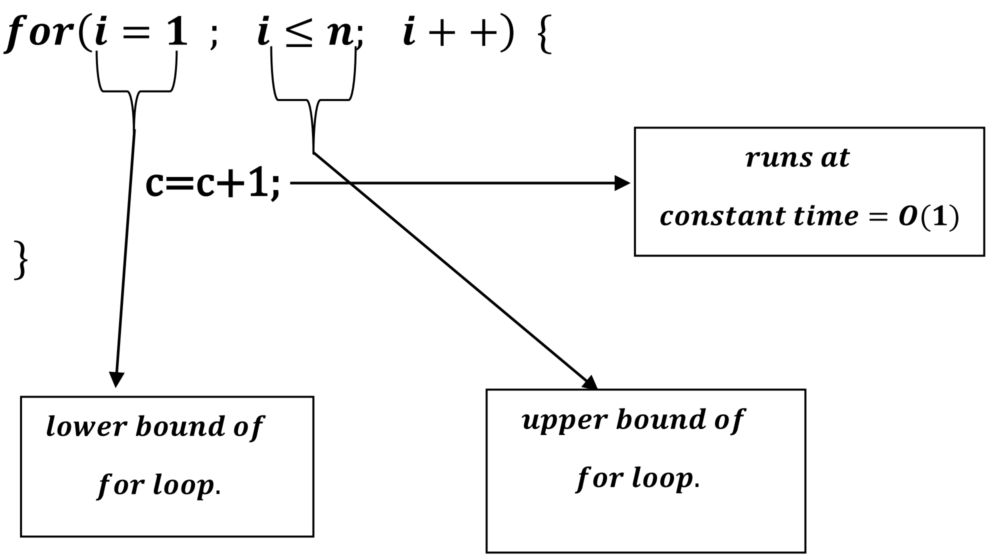
20.ASYMPTOTIC ANALYSIS FOR LOOP.

Approach:

Finding $\boldsymbol{Big\ (O)\ i.e.upto\ n\ times\ run\ of\ the\ particular\ code\ or\ we\ can\ tell\ traverse\ to\ the\ last}$.

When we write for loop, we write it as:



$\boldsymbol{for(i = 1\ ;\ \ i \le n;\ \ i++)\ \{}$

c=c+1;

$\}$

$\boldsymbol{runs\ at\ constant\ time = O(1)}$

$\boldsymbol{lower\ bound\ of\ for\ loop.}$

$\boldsymbol{upper\ bound\ of\ for\ loop.}$

$\boldsymbol{Hence\ `n`\ is\ upper\ bound\ and\ lower\ bound\ is\ 1\ ,input\ size\ is\ 1\ to\ n\ for\ `for\ loop`.}$

Rule 1: How much time the inner most loop's statement run = Time complexity of the loop.

SINGLE LOOP STATEMENT

```
for(i = 1; i ≤ n; i + +){
c = c + 1;
}
```

SOLUTION:

1. *Inner most loop's statement* $\Rightarrow c = c + 1$ *which runs at* $O(1)$ *time.*

2. *No. of inputs in for loop takes* 1 *to* n *times.*
Hence we can say:

lower bound or $c_1 g(n) = 1$, *hence* $g(n) = 1$
upper bound or $c_2 g(n) = n$, *hence* $g(n) = n$

$we\ can\ write\ it\ as : c_1 g(n) \leq f(n) \leq c_2\big(g(n)\big)$

$or, c_1 \times 1 \leq f(1,2,3,\dots,n) \leq c_2 \times n$

$As\ we\ are\ focused\ on\ upper\ bound\ we\ need:$

$f(1,2,3,\dots,n) \leq c_2 \times n\ or\ f(1,2,3,\dots,n) \leq c \times n$

$Hence:$

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

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

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

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

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

$c = c + 1\ runs\ \ in\ 1\ unit\ of\ time\ i.e.1\ time.$

……

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

$c = c + 1\ runs\ in\ 1\ unit\ of\ time\ i.e.1\ time.$

$Rule\ 2: Add\ up\ all\ the\ units\ of\ time\ taken\ by\ innermost\ loop\ statement.$

HENCE, WE ALL KNOW THAT ADDING UP 1 *to* n **TIMES,**

GIVES RESULT $\mathbf{1+1+1+1+\cdots+n=n}$

$$\boldsymbol{Eg: n = 5\,, add\ 1\ to\ 5\ times\ gives\,, 1+1+1+1+1=5}$$

$$\boldsymbol{Eg: n = 6\,, add\ 1\ to\ 6\ times\ gives\,, 1+1+1+1+1+1=6}$$

Hence ,we can tell that :

$$\boldsymbol{Time\ complexity\ of\ the\ \ loop = O\big(g(n)\big)\ = O(n), where}$$

$$\boldsymbol{g(n) is\ the\ upper\ bound\ of\ the\ loop.}$$

****