20.ASYMPTOTIC ANALYSIS FOR LOOP.

Approach:

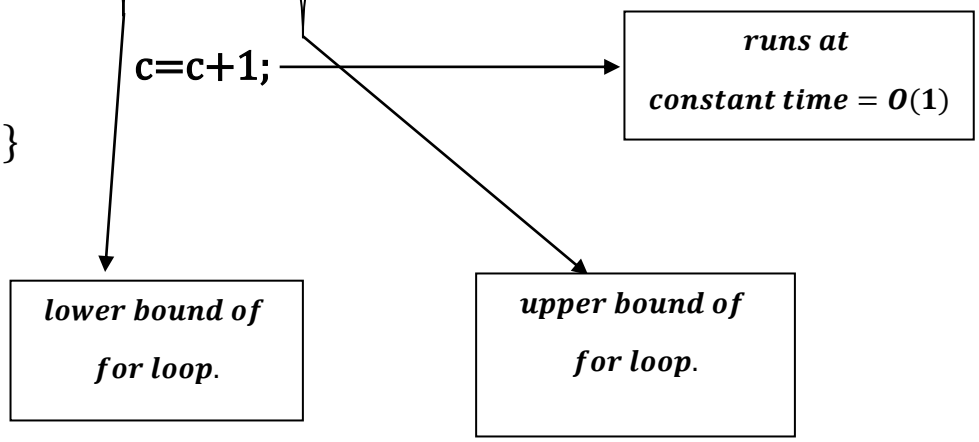
Find*ing Big* (O) *i.e. upto n times run of the particular code or we can tell traverse to the last* .

When we write for loop, we write it as:

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

c=c+1;

}



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 FOR LOOP STATEMENT

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

SOLUTION:

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

2. *No. of inputs in for loop takes* 1 *to* n *times.*

***Hence we can say*:**

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

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

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

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

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

$$Hence:$$

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

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

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

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

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

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

……

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

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

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

$Eg: n = 5$, add 1 to 5 $times$ $gives$, $1 + 1 + 1 + 1 + 1 = 5$

$Eg: n = 6$, add 1 to 6 $times$ $gives$, $1 + 1 + 1 + 1 + 1 + 1 = 6$

$Hence$: we get $O(n)time$ $complexity$.