20.1. ASYMPTOTIC ANALYSIS

NESTED FOR LOOP.

Approach:

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

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

SOLUTION:

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

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

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

$inner\ loop's\ statement.$

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

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

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

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

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

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

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

$\dots \dots \dots.$

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

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

$i.e. when\ i = 1\ , the\ inner\ most\ loop\ statement$

$run\ (1 + 1 + 1 + 1 + \cdots + n) = n\ times$

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

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

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

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

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

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

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

$\dots \dots \dots.$

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

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

$$i.e. when \; i = 2 \; , the \; inner \; most \; loop \; statement$$

$$run \; (1 + 1 + 1 + 1 + \cdots + n) = n \; times$$

.....

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

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

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

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

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

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

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

$$\dots \dots \dots.$$

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

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

$$i.e. when \; i = n \; , the \; inner \; most \; loop \; statement$$

$$run \; (1 + 1 + 1 + 1 + \cdots + n) = n \; times$$

We can add n to n times $[n + n + n + \cdots + n]$gives n^2.

$$Eg: if \; n = 3 \;\; \Rightarrow if \; we \; add \; 3 \; times \; 3 = 3 + 3 + 3 = 9 = 3^2$$