20.10. TIME COMPLEXITY CALCULATION
NESTED FOR LOOP (EG-9).

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

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

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

$\quad for(j = 1; j \leq n - k; j++)\{$

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

$\}$

SOLUTION:

$1. Inner\ most\ loop's\ statement \Rightarrow k = k + 1\ which\ runs\ at$

$O(1)\ time.$

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

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

$inner\ loop's\ statement.$

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

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

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

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

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

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

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

$\dots \dots \dots.$

$f(n-k) \leq c \times n \Longrightarrow when\ j = n-k$

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

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

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

$$T(n) = \sum_{j=1}^{n-k} 1 = (1+1+1+1+\cdots+n-k) = n-k\ times$$

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

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

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

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

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

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

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

$\dots \dots \dots.$

$f(n-k) \leq c \times n \Longrightarrow when\ j = n-k$

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

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

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

$$T(n) = \sum_{j=1}^{n-k} 1 = (1+1+1+1+\cdots+n-k) = n-k\ times$$

.....

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

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

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

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

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

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

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

... ...

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

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

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

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

$$T(n) = \sum_{j=1}^{n-k} 1 = (1 + 1 + 1 + 1 + \cdots + n - k) = n - k\ times$$

$$T(n) = \sum_{i=1}^{n-k} 1 \times \sum_{i=1}^{n-k} 1 = \sum_{i=1}^{n-k} i = (n-k) + (\ n-k) + \cdots n\ times = n(n-k)$$

$$Therefore\ printing\ the\ inner\ most\ statement(k = k + 1)$$

$$n\ times\ n - k = (n-k) + (\ n-k) + \cdots + (n-k) = n(n-k)$$

$$= n^2 - kn\ times\ , hence\ O(n^2 - kn) = O(n^2)$$