19. ANALYSIS OF CONSTANT VARIABLE IN A PROGRAM.

TILL NOW WE HAVE LEARNT THAT:

Big – Oh or Worst Case Complexity:

$$f(n) \leq cg(n), where\ c\ and\ n_0\ are\ constants\ and\ n \geq n_0, then\ f(n) = O(g(n))$$

Big – Omega or Best Case Complexity:

$$f(n) \geq cg(n), where\ c\ and\ n_0\ are\ constants\ and\ n \geq n_0, then\ f(n) = \Omega(g(n))$$

Big – Theta or Average Case Complexity:

$$c_1 g(n) \leq f(n) \leq c_2 g(n), where\ c\ , n_1\ and\ n_2\ are\ constants\ and\ n \geq \{n_1, n_2\},$$

$$then\ f(n) = \Theta(g(n))$$

APPROACH TO FIND TIME COMPLEXITY

Approach to find time complexity of a program is to find the worst complexity i.e., to find $'n'$ times run of a particular code of a program.

FINDING TIME COMPLEXITY OF CONSTANT TERMS.

Suppose in Java we have line of code:

class example {

int a=1;→ $constant\ term$

float b= 2.9f;→ $constant\ term$

char c= 'a;' → $constant\ term$

.....

}

REWRITING IT IN ASYMPTOTIC NOTATION

Therefore we write it as :

1. int a=1; ***It will take one unit of time to run*** . Where n is size of input (no. of input), here no. of input = 1.
   It will be also written as:

   $$f(n) \leq c \times g(n)$$
   $$\Rightarrow f(1) \leq c \times g(1)$$

   $$\Rightarrow \textit{Hence } O(1) \textit{ i.e. } O(c) = O(1), \textit{where c is any constant.}$$

   i.e. It will take one unit of time to run .

2. float b= 2.9f; ***It will take one unit of time to run*** .
   Where n is size of input (no. of input), here no. of input = 1.
   It will be also written as:

   $$f(n) \leq c \times g(n)$$
   $$\Rightarrow f(1) \leq c \times g(1)$$

   $$\Rightarrow \textit{Hence } O(1) \textit{ i.e. } O(c) = O(1), \textit{where c is any constant.}$$

   i.e. It will take one unit of time to run .

3. char c= ‘a;’. Where n is size of input (no. of input), here no. of input = 1.
It will be also written as:

$$f(n) \leq c \times g(n)$$
$$\Rightarrow f(1) \leq c \times g(1)$$

$$\Rightarrow Hence\ O(1)\ i.e.\ O(c) = O(1), where\ c\ is\ any\ constant.$$

$$i.e.\ It\ will\ take\ one\ unit\ of\ time\ to\ run\ .$$

i.e., If there is 165 input size i.e., constant then:

$$f(165) \leq 165 \times 1, here\ constant\ c = 165\ and\ g(n) = 1.$$

$$Then\ O\big(g(n)\big) = O(1).$$

$$f(166) \leq 166 \times 1, here\ constant\ c = 166\ and\ g(n) = 1.$$

$$Then\ O\big(g(n)\big) = O(1).$$

THAT IS FOR ANY CONSTANT VARIABLE IT WILL RUN $O(1)$ $times$.