27. CALCULATION OF TIME COMPLEXITY OF A GIVEN PROGRAM.

$class\ bike$ {

$int\ a; ------\rightarrow c_0\ or\ O(1)$

$int\ b; ---------\rightarrow c_1\ or\ O(1)$

$public\ static\ void\ main(String\ args[])$ {

$bike\ b = new\ bike(); ---\rightarrow c_2\ or\ O(1)$

$b.a = 1; ---\rightarrow c_3\ or\ O(1)$

$b.b = 2; ---\rightarrow c_4\ or\ O(1)$

$int\ n\ = 10; ---\rightarrow c_5\ or\ O(1)$

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

$System.out.println(i); ---\rightarrow c_6n\ or\ O(n)$

}

}

}

Time Complexity:

$f(n) = O(1) + O(1) + O(1) + O(1) + O(1) + O(1) + O(n) = O(n)$

or

$f(n) = c_0 + c_1 + c_2 + c_3 + c_4 + c_5 + c_6n = O(n)$

Example 2

$class\ bike$ {

$int\ a; ------\rightarrow c_0\ or\ O(1)$

$int\ b; ---------\rightarrow c_1\ or\ O(1)$

$public\ static\ void\ main(String\ args[])$ {

$bike\ b = new\ bike(); ---\rightarrow c_2\ or\ O(1)$

$b.a = 1; ---\rightarrow c_3\ or\ O(1)$

$b.b = 2; ---\rightarrow c_4\ or\ O(1)$

$int\ n\ = 10; ---\rightarrow c_5\ or\ O(1)$

$if(b.a == b.b){---\rightarrow c_6\ or\ O(1)}$

$return;$

}

$else{------c_7 or\ O(1)}$

Run n times

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

$System.out.println(i); ---\rightarrow c_8 n\ or\ O(n)$

}

}

}

Time Complexity:

$f(n) = O(1) + O(1) + O(1) + O(1) + O(1) + O(1) + O(1) + O(1) + O(n) = O(n)$

or

$f(n) = c_0 + c_1 + c_2 + c_3 + c_4 + c_5 + c_6 + (c_7 + c_8) \times n = O(n)$

Note: Here we are finding the tight upper bound i.e. running upto `n` times without constant times that doesnot mean the 1 unit of time is not considerable. Hence, greater the lines of code , more the time comsumption in reality.