**Time Complexity**

Relation between input size & running time (operations)

***Example:***

***```***

*Public static void main(String args[]){*

*Scanner sc = new Scanner(System.in);*

*Int n = sc.nextInt();*

*for(int i = 0; i<n; i++){*

*System.out.println(“hello”);*

*}*

*}*

```

Time complexity is directly proportional to input n, so it is a linear relation.

We find time complexity by 3 ways:

1. **Best Case Time Complexity:** Less time required; one operation or one unit time. Numbers – 1,2,3,4,5 ; we have to search 1 then we will find it in 1st position so only 1 operation is required. O(1) *(Omega of 1)*
2. **Average Case Time Complexity:** Average timerequired; Numbers – 2,3,6,4,1,5,7 ; we have to search 1 so this much time is required ((n+1)/2). *Theta((n-1)/2)*
3. **Worst Case Time Complexity:** More time required; we have to perform n operations to get the end result; O(n). *(Big O of n)*

Nested loops – time complexity multiplies (n\*n)

Non nested loops or separate loops – time complexity adds (n + m)

**Space Complexity**

It is the amount of space which a code takes in the memory; it depends on how many variables store in the memory.