Time Complexity

Amount of time taken by algorithm to run as a function of length of input.

```
for(int i=0; i<N; i++)
{
      cout<<"Hello";
                              O(n)
 }
```

Note: It is not actual time, but CPU oper^{n}s.

SPACE COMPLEXITY :-

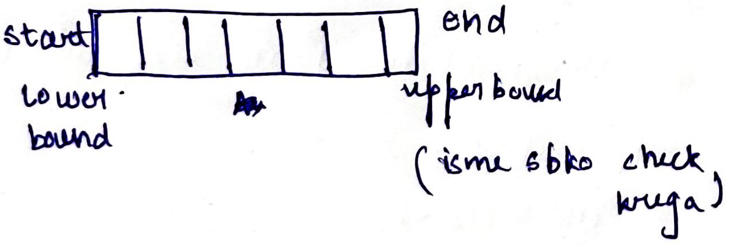
Amount of space taken by algorithm to run as a function of length of input.

Units to represent :-

(i) Big O: Upper bound (Best)

(ii) Theata θ : Average case

(iii) Omega Ω: Lower bound



Big O: complexities:-

1. constant time : O(1)
2. Linear time : O(n)
3. Logarithmic time : O(log N)
4. Quadratic time : O(N^2)

cubic time : O(N^3)

Time complexity order:-

O(1) —— Least complex

O(log N)

O($\sqrt{N}$)

O(N)

O(nlogn)

O(n^2)

O(n^3)

O(2^n)

O(N!)

O(N^n) —— Most complex