Week-3

classmate
Date
Page 62

Time and space complexity

What is Time complexity?
— Amount of Time taken by an algorithm to run as a function of length of input.

What is space complexity?
— Amount of space taken by an algorithm to run as a function of length of input.

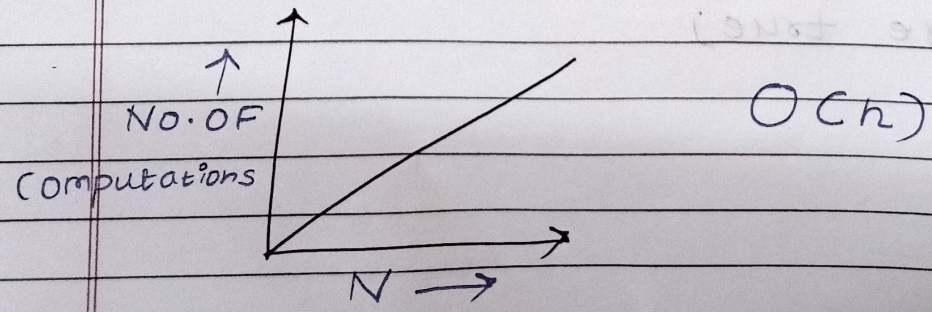
Unit to represent complexity.

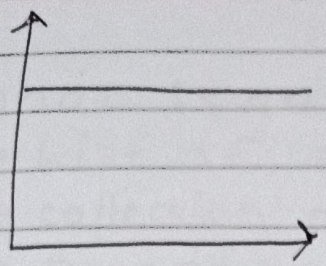
1. Big O : Upper bound
2. Theta Θ : Average bound
3. Omega Ω : Lower bound

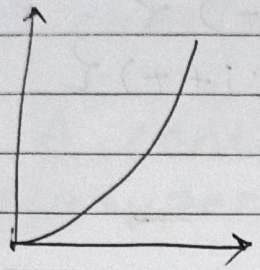
Big O : complexities

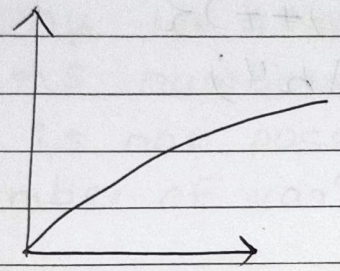
1. constant time : O(1)
2. Linear time : O(N)
   e.g → for (i → N)
3. Logarithmetic time : O(logN)
4. Quadratic time : O(N^2)
5. Cubic time : O(N^3)

Graphs









Least to most complexity

Least

O(1)
O(log N)
O(√N)
O(N)
O(Nlogn)
O(n²)
O(n³)
O(2ⁿ)
O(N!)
O(Nⁿ)

MOST

Example:

```
int main(){
    int a=0, b=0, n;
    cin>>n;
for(int i=0; i<n; i++){
    for(int j=0; j<n; j++){
    cout<<"Hi 1\n";
    }
}
for(int i=0; i<n; i++){
    cout<<"Hi 2\n";
}
return 0;
}
```

T.C → O(N²) + O(N)

Which means,

T.C → O(N²)