21/05/15

(Lec no 11, 12)

Asymptotic Complexities

- The cost of time and space is called Asymptotic Complexities

:- Notations for Asymptotic Complexities :-

i) Big-oh "O" → gives us upper bound
ii) Sigma "Ω" → gives us lower bound
iii) Theta "Θ" → gives us both bounds.

Example-

Run-time for following algorithms are :-

i) a + b → will run in constant time 'c'
ii) c = a + b → 'c'
iii) for (——, ——, —) → 'n'
{
}
iv) n = xyz → 'c'

Total runtime : $T(n) = c + c + n + c$

$= n + c$

i) Big-oh

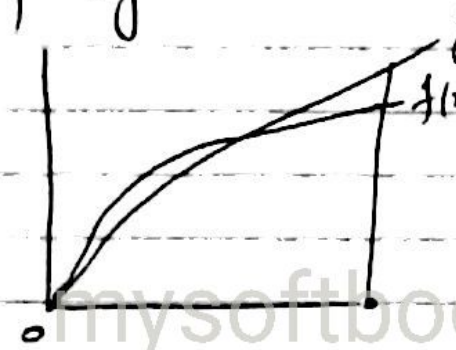
Big-oh : Complexity = $C_{(n)} = n + n^2 + c$

$= O(n^2)$

∴ O will always with biggest value of equation.

Equation:- $\boxed{f(x) \leq cg(x)}$

Graph of Big-oh.-



• A point will come when f(x) will always less than g(x).

ii) Sigma-

Time complexity = $n + n^2 + c$

$= \Omega(n)$

Equation- $\boxed{f(x) \geq c\ g(x)}$ for some n_0

iii) Theta-

Equation-

$$\boxed{f(x) \leq c\ g(x) \leq f(x)}$$

—: Bubble Sort :—

Program Example:-

```
#include <iostream>
#include <conio.h>
using namespace std;
int main ( )
{
    int hold, array [5] ;
    cout << " Enter five numbers " ;
    for ( int i=0 ; i<5 ; i++ )
    {
        cout << " Enter element " ;
        cin >> array [i] ;
    }

    for (int i=0 ; i<4 ; i++)
    {
      for (int j=0 ; j<4 ; j++)
      {
          if ( array [j] > array ( j+1 ] )
          {
              hold = array [j] ;
              array [j] = array [j+1] ;
              array [j+1] = hold ;
          }
      }
}
```

```
cout << " Elements have been Sorted ";
for (int a=0 , a<5 ; a++)
{
    cout << array[a] << endl;
}

return 0;
}
```

— Selection Sort —

```
#include <iostream>
#include <conio.h>
using namespace std;
int main ( )
{
    int size;
    cout << " Enter size of Array ";
    cin >> size;
    int data[size];
    int temp;
    for ( int r=0 ; r<size ; r++)
    {
        cout << " Enter the value ";
        cin >> data[r];
    }
```

```
cout << " Elements in Array are ";
for (int j=0 ; j < size ; j++)
{
    cout << data[j] ; << " " ;
}

for (int i=0 ; i<size ; i++)
{
    for (int h = i+1 ; h < size ; h++)
    {
        if ( data [i] > data [h] )
        {
            temp = data [i] ;
            data[i] = data [h] ;
            data[h] = data [i] ;
        }
    }
}

cout << " Elements have been sorted " ;
for ( int n=0 ; n < size ; n++)
{
    cout << data [n] << " " ;
}

return 0 ;

}
```