# Analysis Of Algorithms.

Eg. Problem : Sum of n natural numbers

Input : $n = 3$

Output : 6      //   $1 + 2 + 3$

Soln 1 :

```
int fun1 (int n)
{
    return n * (n+1)/2 ;
}
```

Soln 2 :

```
int fun2 (int n)
{
    int sum = 0;
    for (int i = 0; i <= n; i++)
        sum = sum + i;
    return sum;
}
```

Solu 3 :

```
int fun3 (int n)
{
    int sum = 0;
    for (int i = 1; i <= n; i++)
        for (int j = 1; j <= i; j++)
            sum++;
    return sum;
}
```

# Asymptotic Analysis

→ The idea is to measure order of growth.

→ Does not dependt upon machine, programming language etc.

→ No need to implement, we can analyze algorithms

Time taken by Solution 1 → $C_1$
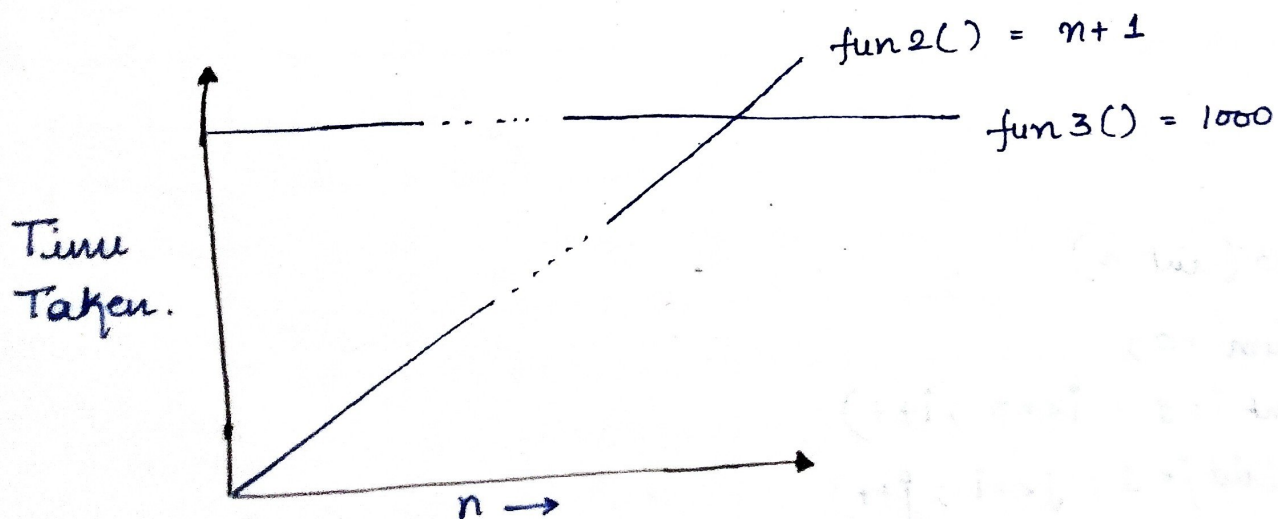
Time taken by Solution 2 → $C_2 n + C_3$

Time taken by Solution 3 → $C_4 n^2 + C_5 n + C_6$

## Order of growth.

→ We can simply ignore the lower order terms and take the higher order term into consideration

For example:
$$C_2 n + C_3 \rightarrow linear$$

fun2() = $n + 1$

fun3() = $1000$

Time Taken.

$n \rightarrow$

fun2() → linear

fun3() → Constant

* After certain value of n fun2() will always take more time than fun3().