**--->Asymptotic Analysis**

This analysis is basically done to find out the order of growth

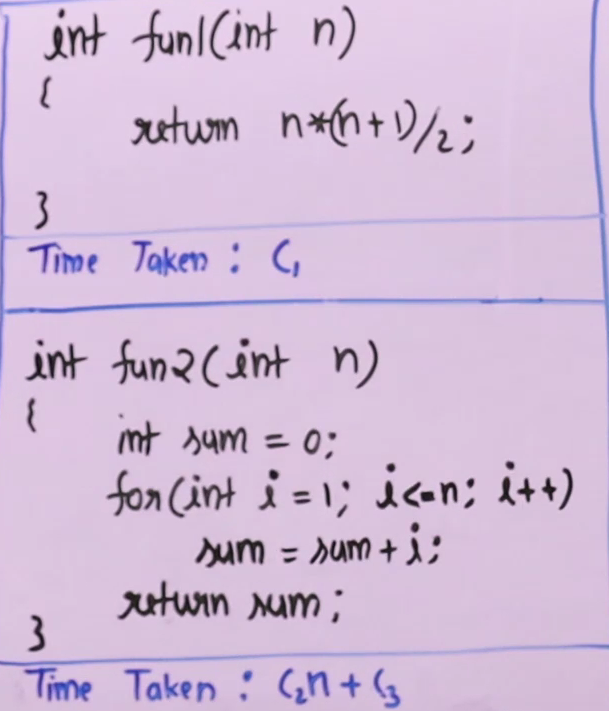
of a program.

The best part is, it does not depend on any programming language

It is algorithm specific not test case specific

Here no need to implement the algorithm, we have to analyse it.

Let us take and example now to get an idea of Asymptotic analysis.

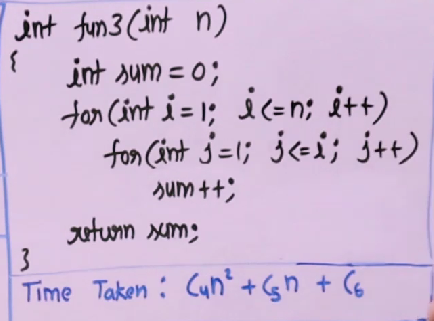


Here we are performing 3 operations

Assuming that the time taken for any value of n time taken is same and constant

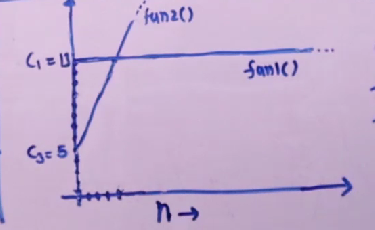
So that’s why time taken is written as C

Some things executing n times and some things at constant n



Similarly, since the loop is executing n^2 time so the time is in quadratic

So, the idea of growth is: Leading term of time taken



Example C1=13

C2=2 and C3=5

Fun1()=13

Fun2()=2n+5

Here after the intersecting value the fun2 grows linearly but fun1 is constant, that’s is order of growth.

In asymptotic analysis we figured out that something which has higher order of growth we will have a higher value after certain value of n.

**---> Example 1:**

Simple function with same order of growth for every input

int getSum(int n, int arr[])

{

int sum=0;

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

{

sum=sum+arr[i];

}

return sum;

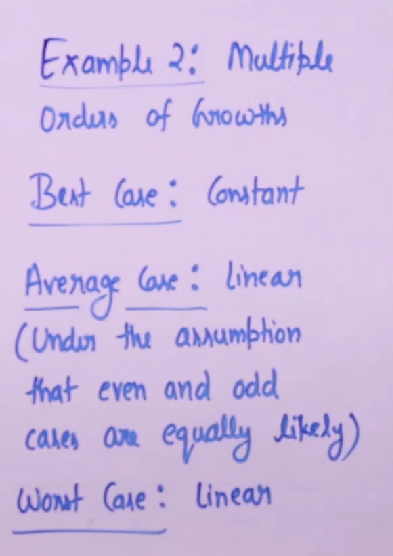
}

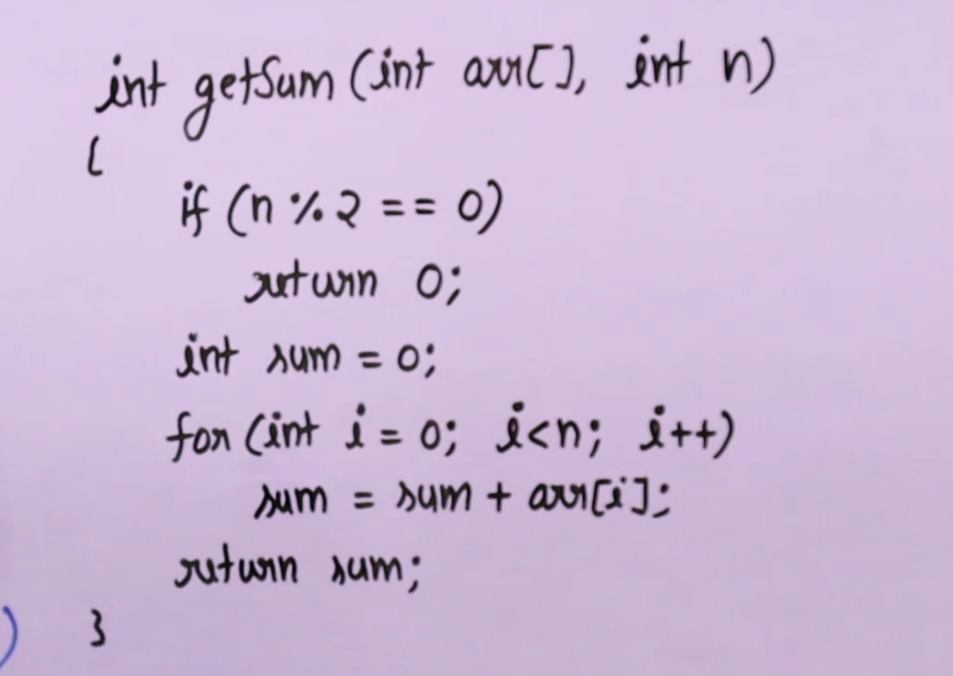
## n is size of array

Time taken by function is C1n+C2

Order of Growth is n(linear)

**----> Example 2:**





Here we can see that for different input we have different time, so we divide the algorithm into 3 cases best, average and worst case

**Best case: Minimum order of growth**

**Average case: The normal average thing of many numbers of inputs by making assumptions**

**Worst case: Linear**

The average case is not done generally, even the best case is not used

In companies the worst case is provided.

:)