DSA 90 Days Plan notes

(Starting – 26th Oct)

**Starting with Day 1; date: 27 Oct:**

Starting recursion with love Babbar…

Recursion – time complexity – O(nn)

What is recursion?

When a function calls itself within its self-body it is called as recursion.

Or

When a big problem depends upon same type of small problems to solve can be solved by recursion

E.g. –

If we take the example of 2n

2n = 2 \* 2n-1

So here 2n & 2n-1 are same type of sub problems & can be solved by recursion

Here the recurrence relation is

F(n) = 2 \* f(n-1)

E.g. –

If we take the factorial also it is a recurrence function

5! = 5 \* 4 \* 3 \* 2 \* 1

5! = 5 \* 4!

So there is also recurrence applied

F(n) = n \* f(n-1)

We need a base case to solve a problem in recurrence problem base case decides when we must stop the recursion.

In the base case the return is compulsory.

Factorial Code - [Check the code file…](Codes.cpp)

If there is not a base case, then there is a segmentation fault. Because of the infinity time running of the recursive function and till the memory stack of the program is not full.

It works as a call stack and return the value to the parent function and chaining likely

#include <bits/stdc++.h>

using namespace std;

int **factorial**(int n){

    return n==0?1:n\***factorial**(n-1);

}

int **main**(){

    //Lets Make A Better World

    int a = 0;

    cin**>>**a;

    cout**<<factorial**(a);

    return 0;

}