#Recursion

Why Stackoverflow!

What is Stackoverflow?

#Factorial

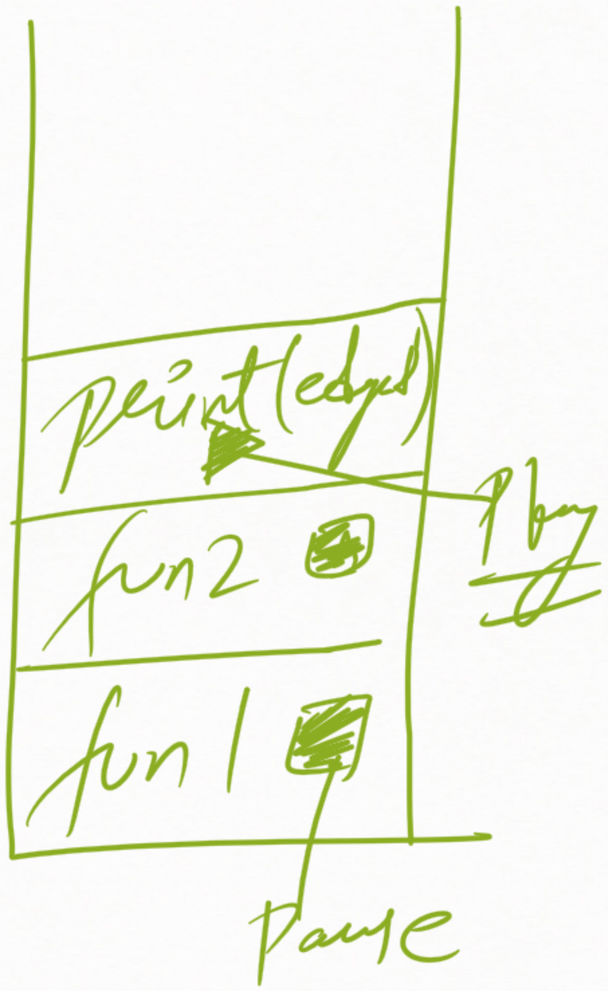
$$n! = n(n-1)(n-2)\ldots\ldots 1$$

$$n! = \begin{cases} n(n-1)! & (\text{if } n > 0) \\ 1 & (\text{if } n = 0) \end{cases}$$

```
fun factorial(n)
{
    if n==0
        return 1
    else
        return n*factorial(n-1)
}
```

```
def fun2()
{
    print(edyoda)
}

def fun1()
{
    print(happy)
    fun2()
    print(students)
}
fun1()
```

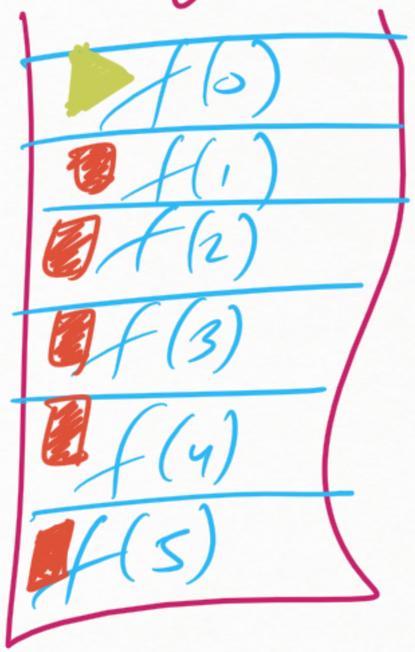


Def^n of Recursion

=) Function calling itself

=) Actually function calling an copy of itself.

```
factorial(n)
{
    if n == 0
        return 1
    else return n * factorial(n-1)
}
```



→ Stack with elements getting full.

→ So when stack gets full with recursion calls, stackoverflow happens

→ Error: Maximum recursion depth exceeded.

→ Stack becomes full, more memory needed, Program Crashes.

Fibonacci Numbers without recursion

① Method

Fib(n)

{ if (n <= 1)

return n

$F_1 = 0$

$F_2 = 0$

for i = 2 to n

$F = F_1 + F_2$

$F_1 = F_2$

$F_2 = F$ }

Code ↓

```python
# iterative method in python
def fib_2(n):
    if n<=1:
        return n
    else:
        f1 = 0
        f2 = 1
    for i in range(2,n+1):
        f = f1+f2
        f1 = f2
        f2 = f
    return f

print(fib_2(100))
```

② Method to calculate fibonaci with recursion

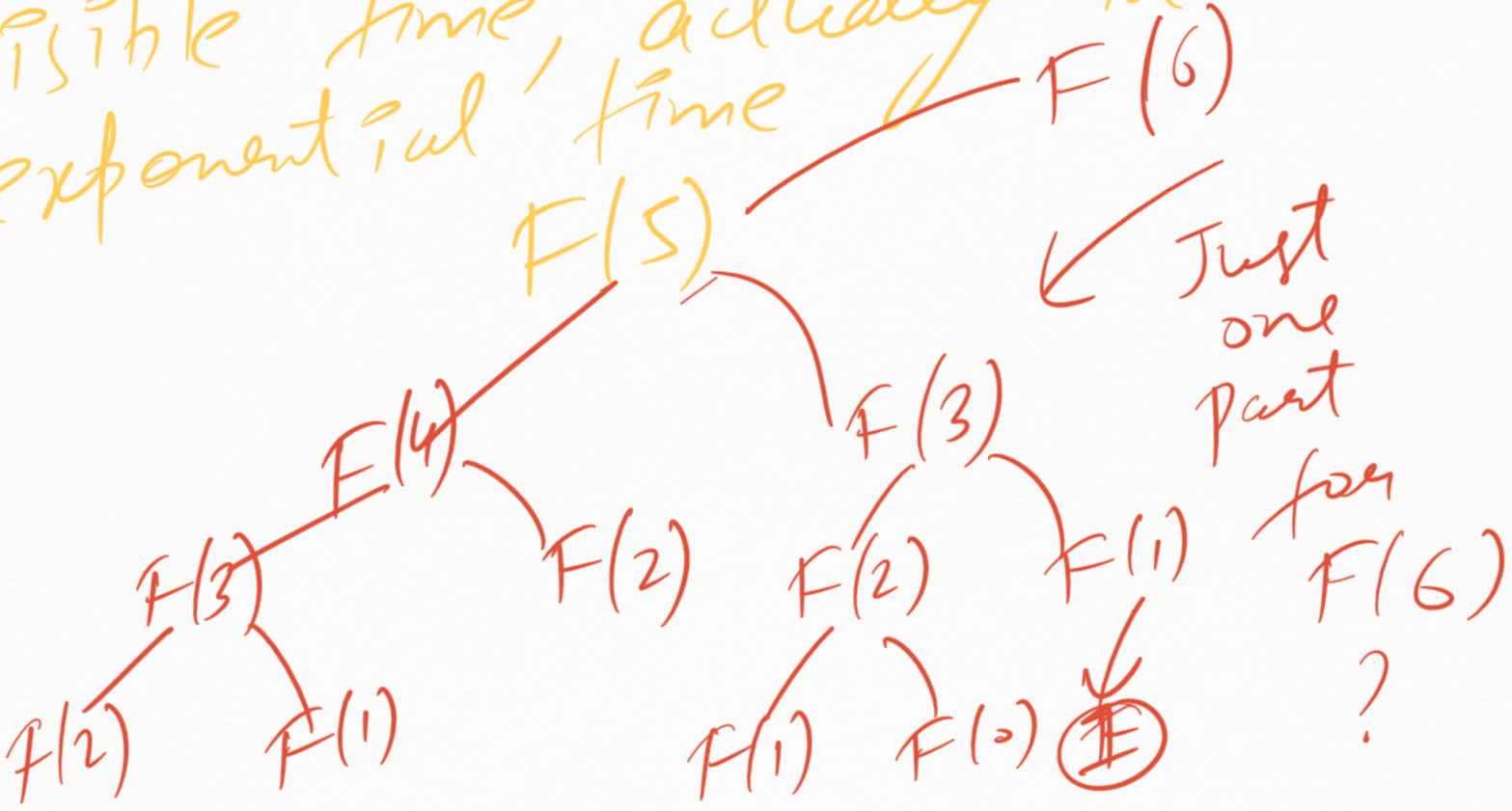
```
fib(n)
{
  if n == 0 or n == 1:
      return n
  else
      return fib(n-1) + fib(n-2)
}
```

Code

```python
# recursive method in python
def fib(n):
    if n==0 or n==1:
        return n
    else:
        return fib(n-1) + fib(n-2)

print(fib(100))
```

2 method does not work for n = 50, 100
in visible time, actually it
takes exponential time



$$T(n) = T(n-1) + T(n-2) + C$$

$$T(n) \simeq 2T(n-1) + C$$

$$T(n-1) \simeq T(n-2) + T(n-3) + C$$

$$T(n) \simeq 2^k T(n-2k) + C$$

$n - k = 0$

$n = k \Rightarrow k = n$

$O(2^n)$

Exponential

Tower of Hanoi

Code ↓

★ Visualization on the internet

```
# Recursive Python function to solve tower of hanoi

def TowerOfHanoi(n , from_rod, to_rod, aux_rod):
    if n == 1:
        print "Move disk 1 from rod",from_rod,"to rod",to_rod
        return
    TowerOfHanoi(n-1, from_rod, aux_rod, to_rod)
    print "Move disk",n,"from rod",from_rod,"to rod",to_rod
    TowerOfHanoi(n-1, aux_rod, to_rod, from_rod)

# Driver code
n = 4
TowerOfHanoi(n, 'A', 'C', 'B')
# A, C, B are the name of rods
```