Recursion:

- functions/methods that call themselves
- problems defined in terms of themselves

factorial

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

---

or

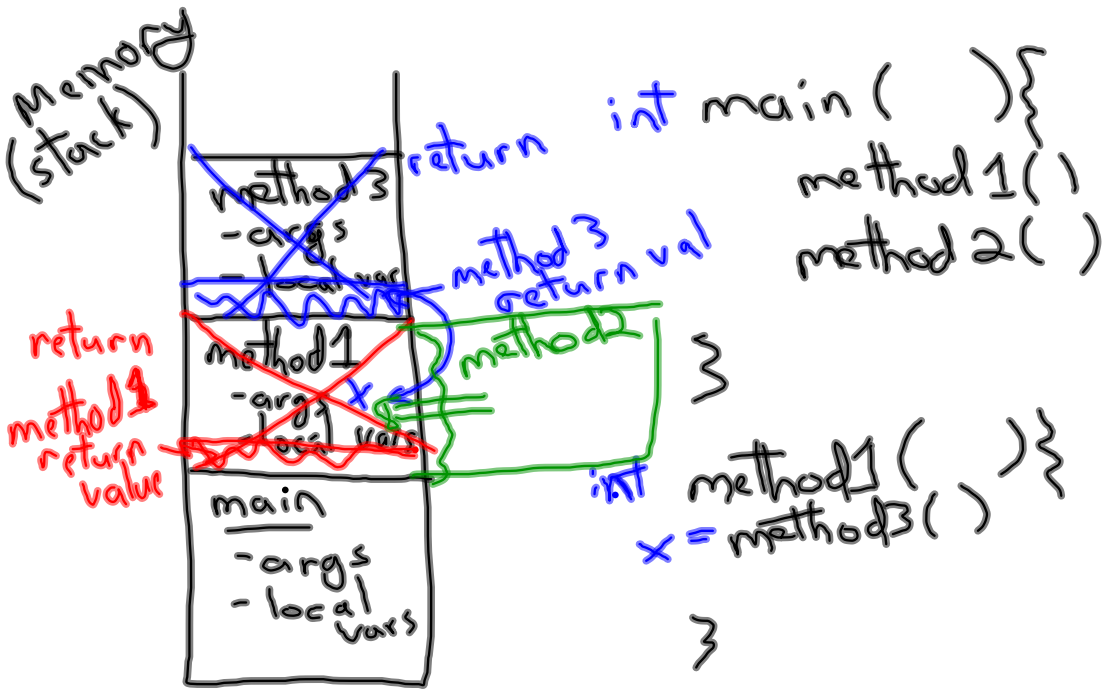
$$n! = \begin{cases} n \cdot (n-1)! & n > 0 \\ 1 & n = 0 \end{cases}$$

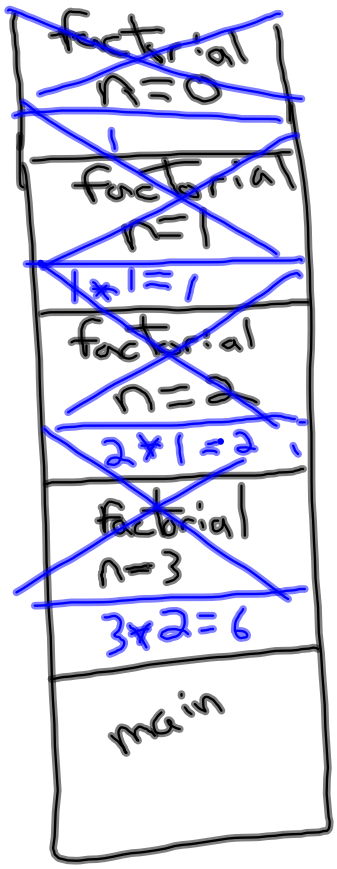
```
public static int factorial(n){
    if (n==0)
        return 1;
    else return n*factorial(n-1);
}
```

base case

function call stack

in memory
- store: parameter values
  local variables
- return value



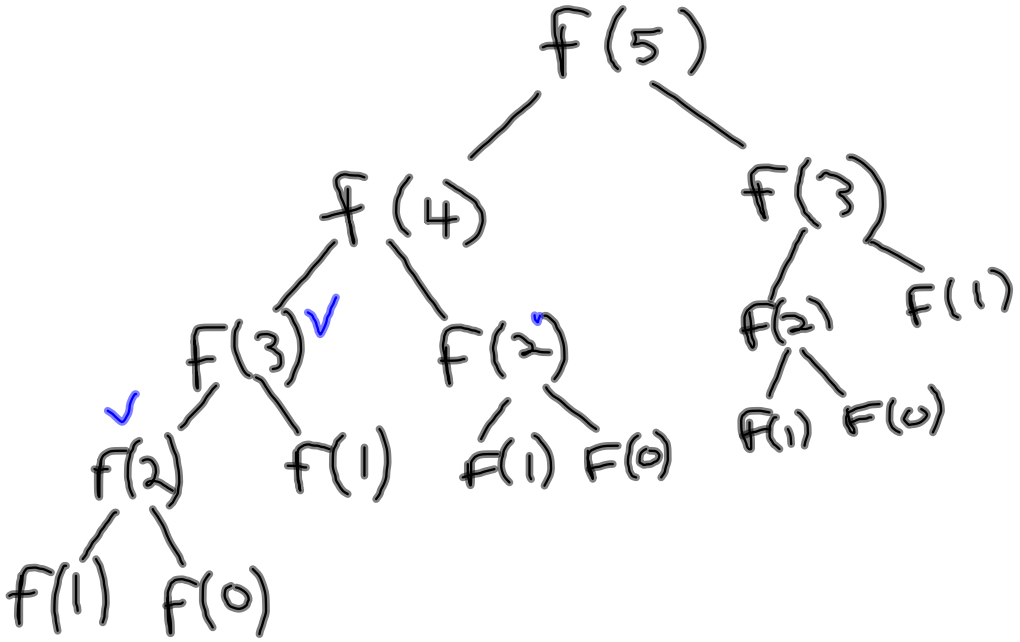

factorial
n=0
1
factorial
n=1
1*1=1
factorial
n=2
2*1=2
factorial
n=3
3*2=6
main

factorial(3)

Fibonacci numbers

$$F(n) = \begin{cases} 0 & n=0 \\ 1 & n=1 \\ F(n-1)+F(n-2) & n>1 \end{cases}$$

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...



Towers of Hanoi