

# Recursion

## Recursion

Miguel Tinta Aguilar, Angelo Figueroa Vega

System Engineering School  
System Engineering and Informatic Department  
Production and Services Faculty  
San Agustín National University of Arequipa

August 2, 2020



Definition

How does it work

Recursion and Iteration

Fibbonaci

References

# What is Recursion

Recursion is a method of solving a problem where the solution depends on solutions to smaller instances of the same problem. To do these tasks the recursion involves a function calling itself.

## Example

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

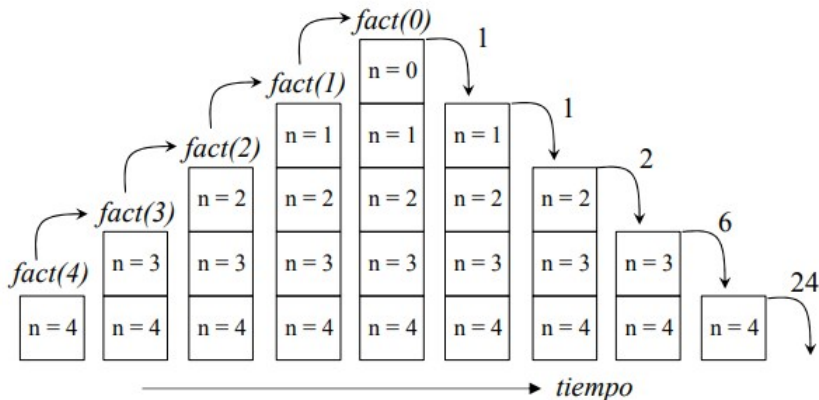
# How Recursive methods works

To write a recursive method you need to know two parts:  
A base case and a general case. The first one will make the recursive method reach an end, and the second one it is the operation.

## Example

```
static int factorial(int n){  
    if (n == 0)  
        return 1;        //BASE CASE  
    else  
        return(n * factorial(n-1)); //GENERAL CASE
```

# Graphic Representation



# Important aspects to be taken into consideration when using recursion and iteration

**The computing load** Time of Ejecution and memory used.

**Redundancy** Sometimes Recursion resolves the same problem multiple times.

**Solution** Sometimes an iterative solution it is too complicate to find

**Resultant code** Using recursion, the final code might be more concise, elegant and easy to read and understand



# Comparison between Recursion and Iteration

## Recursion

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

## Iteration

```
public static int factorial(int n) {  
    if (n == 0) {  
        return 1;  
    }else {  
        int factorial = 1;  
        for(int i=1;i<=n;i++) {  
            factorial = factorial * i;  
        }return factorial;  
    } }  
}
```

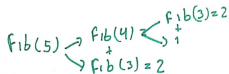
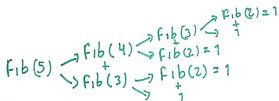
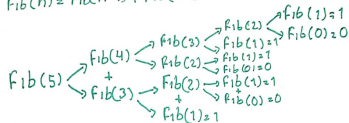
## Recursive Fibonacci sequence

```
public class basic_example1 {  
    public static void main(String[] args){  
  
        System.out.println(fibonacci(x));  
    }  
    public static int fibonacci (int n){  
        if (n<2) {  
            return n;  
        } else {  
            return fibonacci(n-1) + fibonacci(n-2);  
        }  
    }  
}
```

$$\text{Fib}(0) = 0$$

$$\text{Fib}(1) = 1$$

$$\text{Fib}(n) = \text{Fib}(n-1) + \text{Fib}(n-2)$$



$$\text{Fib}(5) \rightarrow \frac{3}{2} > \boxed{5} \quad \boxed{\text{Fib}(5) = 5}$$

## CONCLUSION

- ▶ Recursion is used in programming languages that do not have control structures. For example, functional programming languages.
- ▶ For the elaboration of recursion algorithms a knowledge of mathematical induction is necessary.
- ▶ Recursion is normally used for solving complex problems that would not be solved with iteration or would be very difficult to solve.

## References - Web pages

- ▶ <https://elvex.ugr.es/decsai/java/>
- ▶ <https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html>
- ▶ <https://www.libertaddigital.com/internet/que-es-la-recursividad-que-es-la-recursividad-que-es-l>
- ▶ [https://en.wikipedia.org/wiki/Recursion\\_\(computer\\_science\)](https://en.wikipedia.org/wiki/Recursion_(computer_science))