



1INF27

# Algoritmia y Estructura de Datos

2024

**Profesores:**

Cueva, R.		Allasi, D.		Roncal, A.		Huamán, F.
0581		0582		0583		0584

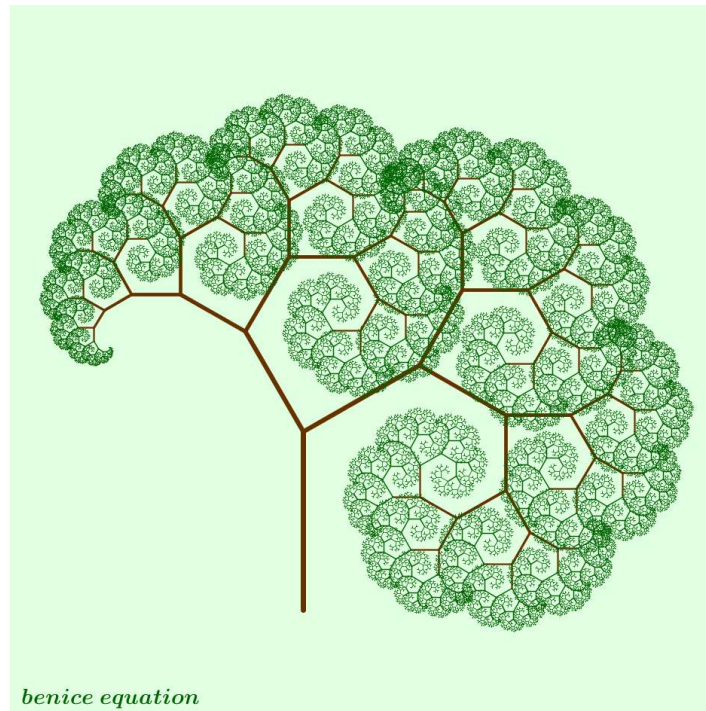
# Capítulo 2

# RECURSIÓN



## RECURSIÓN

- Un objeto es recursivo si podemos definirlo en función de sí mismo.



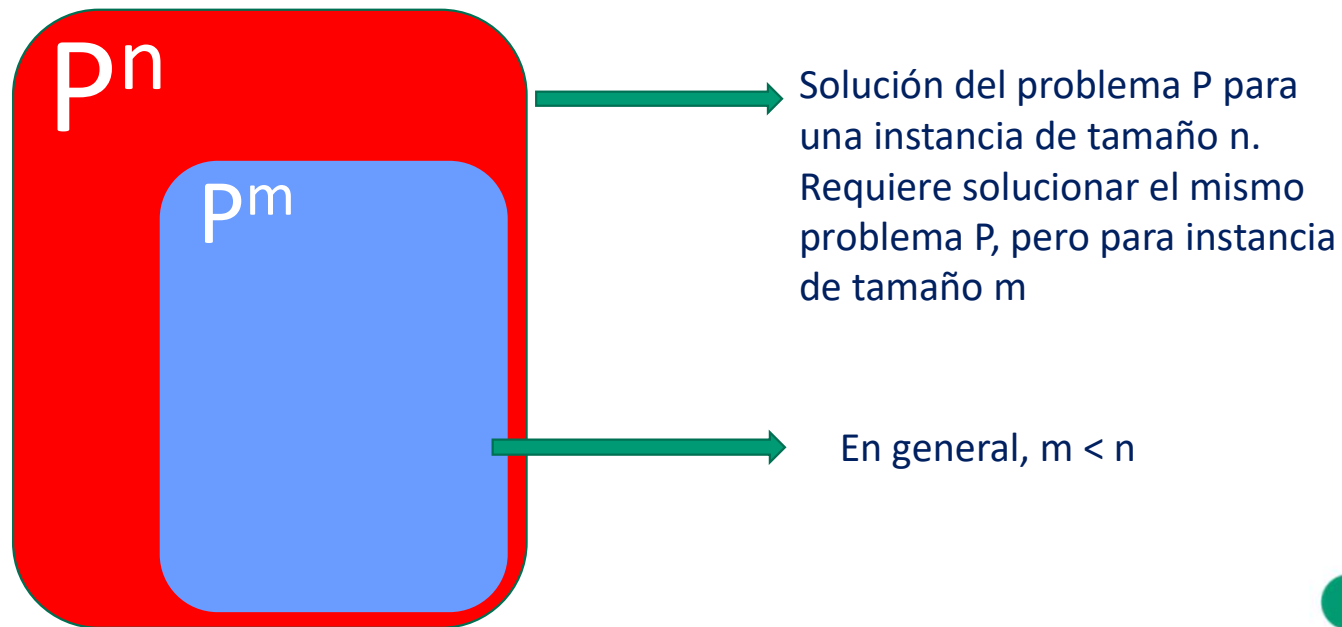
## RECURSIÓN

- Considerando una instancia mas pequeña.



## RECURSIÓN

- En computación, la recursión es útil cuando la solución a un problema puede definirse en términos de la solución del mismo problema, pero sobre una instancia más pequeña.

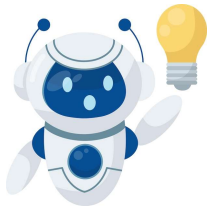


## RECURSIÓN

- Ejemplo 1: Calcular el factorial de un número  $n$
- Por definición

$$n! = n \times (n - 1)! \quad \longrightarrow \quad \text{Caso recursivo}$$

$$0! = 1 \quad \longrightarrow \quad \text{Caso base}$$



*Todo programa recursivo debe tener un caso base, que asegure que el programa termina en algún momento*



## RECURSIÓN

### Consideraciones:

- Los objetos definidos en una función recursiva son locales. Éstos no se transmiten a las llamadas recursivas.
- Aunque las variables y parámetros tienen el mismo nombre, éstos podrían contener diferentes valores.
- Ejemplo factorial



## Recursión - Factorial

- **Pre-condición:** número  $n \geq 0$
- **Post-condición:** factorial de  $n$
- **Funcion** *factorial( $n$ )*
  - Si  $n = 0$*   
*retornar 1*
  - en caso contrario*  
*retornar  $n * \text{factorial}(n - 1)$*





## Recursión - Fibonacci

- Ejemplo 2: Calcular el n-ésimo número de la serie Fibonacci

0, 1, 1, 2, 3, 5, 8, 13, 21, ...

Caso base



$$F(1) = 0$$

$$F(2) = 1$$

Caso recursivo



$$F(n) = F(n - 2) + F(n - 1)$$



## Recursión - Fibonacci

- **Pre-condición:** número  $n \geq 1$
- **Post-condición:**  $n$ -ésimo número de serie Fibonacci
- **Funcion** *Fibonacci( $n$ )*
  - Si  $n = 1$*   
*retornar 0*
  - en caso contrario Si  $n = 2$*   
*retornar 1*
  - en caso contrario*  
*retornar Fibonacci( $n - 1$ ) + Fibonacci( $n - 2$ )*



## Recursión - Ejemplos

- Implementar una función recursiva que permita obtener el mayor número de un arreglo de N números enteros
- Implementar una función recursiva que indique si un número dado se encuentra en un arreglo de N números enteros
- Implementar una función recursiva que reciba como único parámetro un número en base 10 y lo imprima en base 2



## RECURSIÓN

### Características:

- Debe contener siempre algún caso base. Esto garantiza que el programa no corra infinitamente.
- Generalmente un algoritmo recursivo contiene una condición que permite saber si necesitamos detener la recursión o continuar.



## RECURSIÓN

### Consejos:

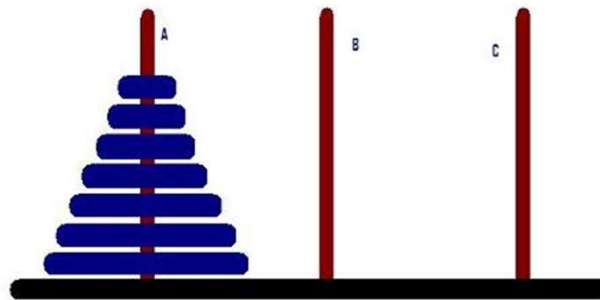
- La recursión se debe emplear sólo cuando el problema se define en términos recursivos
- La mayoría de algoritmos iterativos pueden implementarse de forma recursiva, lo cual no significa que la forma recursiva sea más eficiente.
- Todo algoritmo recursivo puede implementarse de manera iterativa (algunas veces usando técnicas avanzadas de algoritmos).
- Si el problema es recursivo, la implementación recursiva es la mejor opción en términos de claridad.

## RECURSIÓN - Resumen

- Una función es recursiva si se invoca a sí misma.
- Definición de funciones recursivas
  - Siempre definir caso(s) bases
  - Definición de caso recursivo es generalmente la definición del problema en sí
- ¿Recursión o Iteración?
  - Algoritmos recursivos son la mejor opción (por claridad), cuando el problema es recursivo.
  - Algoritmos recursivos son siempre más lentos algoritmos iterativos.

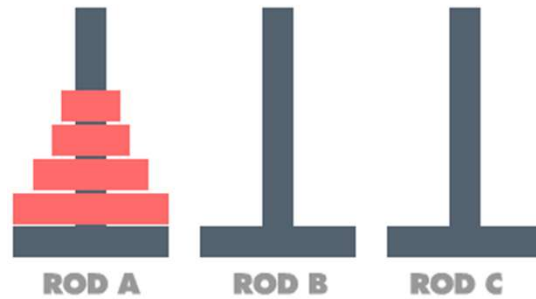
## RECURSIÓN – Torres de Hanoi

- Existen 3 torres en las que se alojan discos, cada uno ligeramente inferior en diámetro al que está justo debajo de él. Se desea determinar los movimientos necesarios para trasladar los discos de una torre a otra cumpliendo las siguientes reglas:
  - Sólo se puede mover un disco a la vez
  - Nunca puede quedar un disco sobre otro de menor tamaño

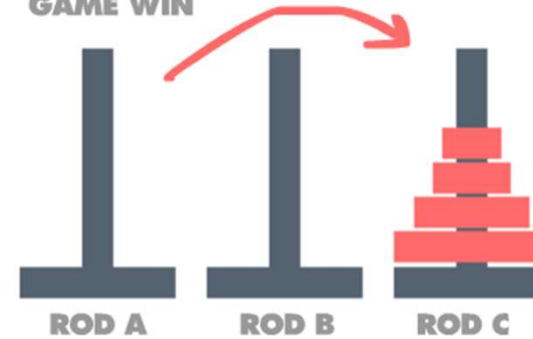


## RECURSIÓN – Torres de Hanoi

GAME START



GAME WIN



VALID MOVE



INVALID MOVE





## RECUSIÓN – Torres de Hanoi



## RECURSIÓN – Torres de Hanoi

```
Funcion  hanoi(n,A,B,C)
  Si n≠0 Entonces
    hanoi(n-1,A,C,B)
    Escribir "Mover disco ",A," a " C
    hanoi(n-1,B,A,C)
  FinSi
FinFuncion
```

## RECURSIÓN – Tarea

- Un palíndromo es una palabra que se escribe exactamente igual , leído en un sentido o en otro. La palabra level es un palíndromo, por ejemplo.
- Escribir una función recursiva que devuelva el valor 1 (verdadero) si una palabra es palíndromo y 0 (falso) en caso contrario.
- ¿Es posible adaptar el problema de las torres de Hanoi para n bases (torres) ?

