SESIÓN **/11**

RECURSIVIDAD

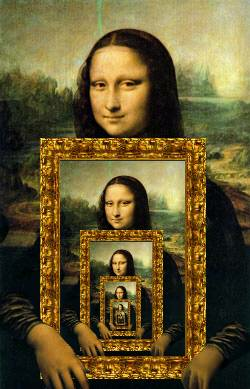
1. Definición
2. Características
3. Algoritmos recursivos
4. Aplicación

**INTRODUCCIÓN**

* La clase anterior conocimos la Estructura de tipo FIFO, llamada también Cola. Describimos los algoritmos para realizar operaciones sobre una Cola y su aplicación en el mundo real.
* En la presente sesión, conoceremos los procedimientos recursivos. Revisaremos sus principales características, la forma de implementarlos y su aplicación en programas para computadora.
* Culminaremos comparando la eficiencia de los procedimientos recursivos versus los procedimientos iterativos.

1. **DEFINICIÓN**

Es una técnica que consiste en definir un proceso en términos de sí mismo. Se utiliza para resolver complejos problemas de programación que, por naturaleza, son repetitivos.



1. **CARACTERÍSTICAS**

Las principales características de la Recursión o Recursividad son las siguientes:

* Se basa en el principio divide y vencerás. Se descompone un problema en versiones más pequeñas, luego se construye una solución para el todo.
* La recursividad se implementa en un programa utilizando un procedimiento recursivo. El procedimiento recursivo es una función ó método que se invoca así mismo.
* Todo procedimiento recursivo tiene una condición por la cual se termina la recursividad. A esta condición se le conoce como criterio base.

***EJEMPLO***

public void procesar ( )

{

…………. ;

procesar ( );

…………. ;

}

***Dada la siguiente función:***

***Dentro de la función procesar se invoca a la función procesar.***

1. **ALGORITMOS RECURSIVOS**
   1. Factorial de un número

El factorial de un número entero n, se define como el resultado de multiplicar los números enteros comprendidos entre 1 y n.

Sea la función factorial, la cual se define de la siguiente manera:

n! = n x (n–1) x (n–2) x …….. x 2 x 1

La misma función puede ser definida, en forma recursiva, de la siguiente manera:

n! = n x (n–1)!

Donde n debe ser un número entero. Además, por definición matemática: 0! = 1

Expresado como una función matemática:

n x factorial (n–1) Si n > 0

factorial (n) =

1 Si n = 0

El factorial de un número entero n es definido con valores más pequeños de n, los cuales se encuentran cada vez más cerca del criterio base (n = 0).

* 1. Máximo común divisor

El máximo común divisor de 2 o más números enteros, es el mayor número entero que los divide sin dejar residuo.

Expresado como una función matemática:

mcd (a–b, b) Si a ≥ b

mcd (a, b–a) Si a < b

mcd (a, b) =

a Si b = 0

b Si a = 0

El máximo común divisor de 2 números enteros (a y b), es definido con valores más pequeños de a y de b, los cuales se encuentran cada vez más cerca del criterio base (a = 0 ó b = 0).

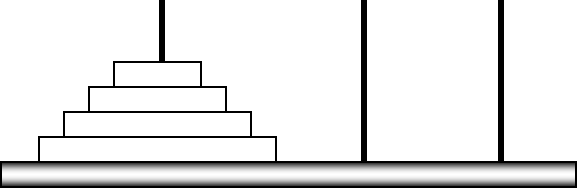
1. **APLICACIÓN**

Uno de los desafíos matemáticos que puede ser resuelto aplicando procedimientos recursivos, son las Torres de Hanoi.

TORRES DE HANOI

Es un juego de mesa, inventado por el matemático francés Edouard Lucas en 1883. Está conformado por n discos y 3 torres (ó postes). Todos los discos se colocan en el primer poste, el disco de mayor tamaño va al fondo y los demás discos se colocan en orden decreciente.

A B C

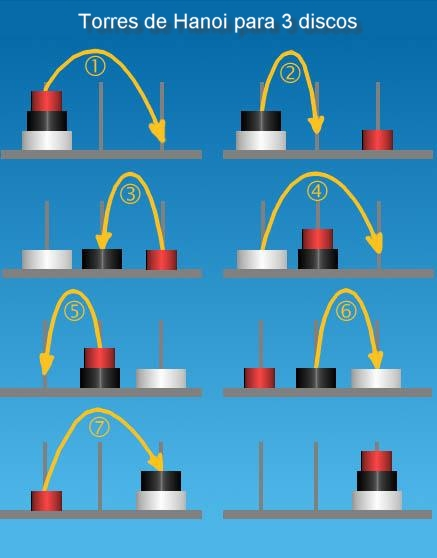


El objetivo del juego es mover todos los discos al tercer poste en el menor número de movimientos, utilizando el segundo poste como intermediario. Además, se debe tener presente las siguientes reglas:

* Sólo se puede mover un disco a la vez.
* Un disco de mayor tamaño no se puede colocar sobre otro disco de menor tamaño.

***EJEMPLO:***

***Descripción de movimientos para trasladar 3 discos del primer al tercer poste.***



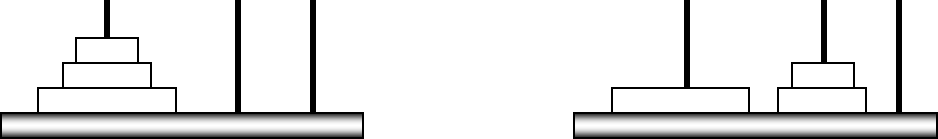
***Fue necesario realizar 7 movimientos para trasladar los 3 discos.***

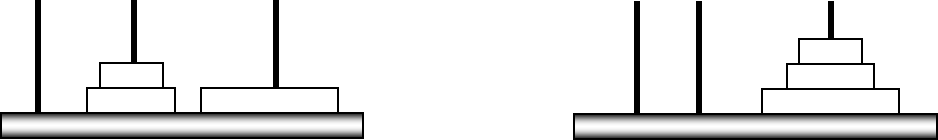
La secuencia de movimientos es la siguiente:

1. Del poste A al poste C.
2. Del poste A al poste B.
3. Del poste C al poste B.
4. Del poste A al poste C.
5. Del poste B al poste A.
6. Del poste B al poste C.
7. Del poste A al poste C.

En general, para trasladar n discos del poste de origen al poste destino, el número mínimo de movimientos que se debe realizar es: 2**n** – 1

Se puede utilizar un algoritmo recursivo para trasladar n discos desde el poste de origen hacia poste destino, utilizando, además, un poste temporal.





1. Mover los n-1 discos superiores del poste de origen al poste temporal.
2. Mover el disco del poste de origen al poste destino.
3. Mover los n-1 discos del poste temporal al poste destino.

**CONCLUSIÓN**

Cada vez que se invoca a un procedimiento recursivo, estamos más cerca del criterio base. Algunos procedimientos recursivos deben devolver un valor al llegar al criterio base. Los procedimientos recursivos no siempre son más eficientes que los procedimientos iterativos.

**REFERENCIAS BIBLIOGRÁFICAS**

* Cairo, O.; Guardati, S. (2008). Estructuras de datos. 3ra. Edición. México D.F., Mexico: McGraw Hill.
* Instituto NIIT (2011). Data Structures and Algorithms. Student guide.