

4.2 EL PROBLEMA DE LAS TORRES DE HANOI

El problema de las Torres de Hanoi es un problema clásico de recursión, ya que pertenece a la clase de problemas cuya solución se simplifica notablemente al utilizar este concepto.

Se tienen tres torres y un conjunto de N discos de diferentes tamaños. Cada uno de ellos tiene una perforación en el centro que le permite deslizarse por cualquiera de las torres. Inicialmente los N discos están ordenados de mayor a menor en una de las torres.

El objetivo del problema consiste en pasar los N discos de la torre de origen a una torre destino, utilizando la otra torre disponible, como auxiliar. A continuación se presentan las reglas que se deben respetar en cada movimiento.

1. En cada movimiento sólo puede intervenir un disco; por lo tanto, siempre será el disco superior el que pueda moverse.
2. No puede quedar un disco sobre otro de menor tamaño.

Supongamos que las torres se identifican con los nombres A , B y C . Los discos inicialmente se encuentran en la torre A —origen—. El objetivo, como señalamos anteriormente, consiste en transferir todos los discos a la torre B —destino—, utilizando la torre C como auxiliar. En las figuras 4.13a y b se presentan el estado inicial y el estado final, respectivamente, del problema de las Torres de Hanoi para cinco discos.

Es importante observar, si se analiza detenidamente el problema, que éste se puede descomponer en tres subproblemas, uno de los cuales, el segundo, se considera trivialidad porque implica únicamente un movimiento. Se muestran a continuación los diferentes subproblemas:

1. Transferir $(N - 1)$ discos de la torre A —origen— a la torre C —auxiliar—.
2. Mover un disco de la torre A —origen— a la torre B —destino—.
3. Transferir $(N - 1)$ discos de la torre C —auxiliar— a la torre B —destino—.

En la figura 4.14 se presenta la solución al problema de las Torres de Hanoi para dos discos, $N = 2$.

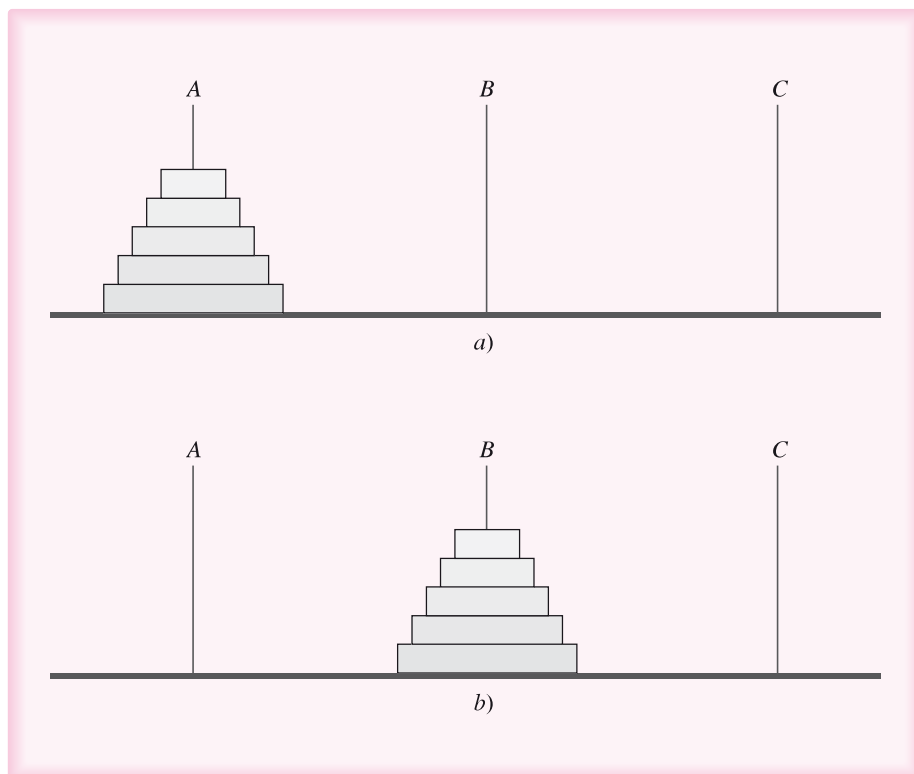
Observe que se realizan tres movimientos:

Mover de A a C
 Mover de A a B
 Mover de C a B

En la figura 4.15, por otra parte, se ilustra la solución del problema de las Torres de Hanoi para tres discos, $N = 3$. Primero se transfieren dos $(N - 1)$ discos de la torre A a la torre auxiliar C (figuras 4.15b, c y d). Posteriormente se realiza el movimiento del disco de la torre A a la torre destino B (figura 4.15e). Finalmente se resuelve el tercer subproblema, se transfieren $(N - 1)$ discos de la torre C —auxiliar— a la torre A (figuras 4.15f, g, h).

FIGURA 4.13

Torres de Hanoi.
a) Estado inicial.
b) Estado final.



Los movimientos que se realizan para resolver este problema son: mover de A a C

Mover de A a B
Mover de B a C
Mover de A a B
Mover de C a A
Mover de C a B
Mover de A a B

Luego de realizar numerosas pruebas para distintos valores de N , se puede concluir que para cualquier N , el número de movimientos (NM) está dado por la siguiente fórmula:

$$NM = 2^n - 1$$

Fórmula 4.10

Así, por ejemplo, para cinco discos se efectuarán 31 movimientos, para diez discos 1 023 movimientos y para 15 discos 32 767 movimientos.

A continuación se presenta el algoritmo recursivo que permite resolver este problema.

FIGURA 4.14

Torres de Hanoi ($N = 2$).
a) Estado inicial. b) Luego de mover un disco de A a C. c) Luego de mover un disco de A a B. d) Estado final.

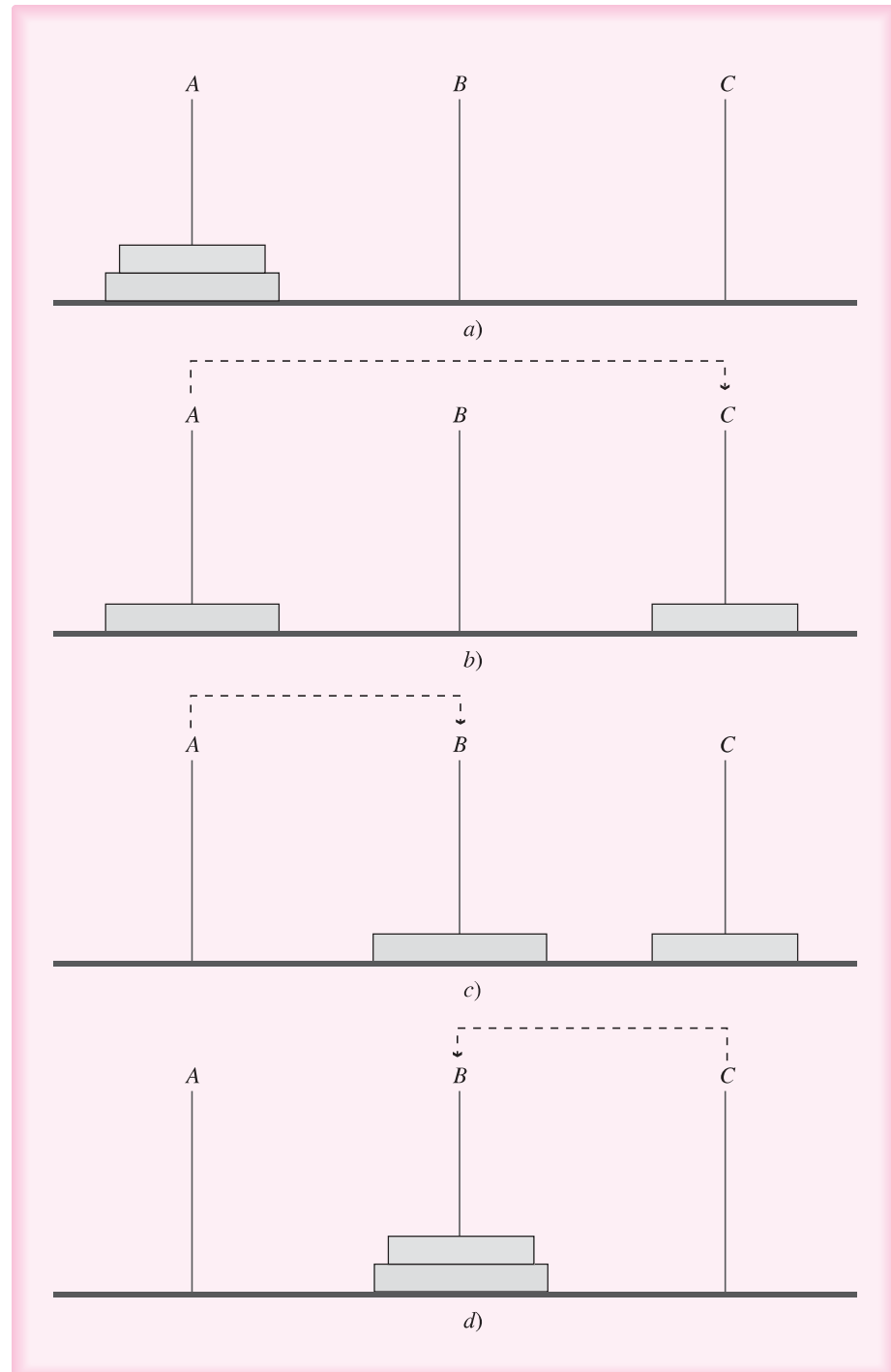


FIGURA 4.15Torres de Hanoi ($N = 3$).

a) Estado inicial.

b) Luego de mover un disco de A a B.

c) Luego de mover un disco de A a C.

d) Luego de mover un disco de B a C.

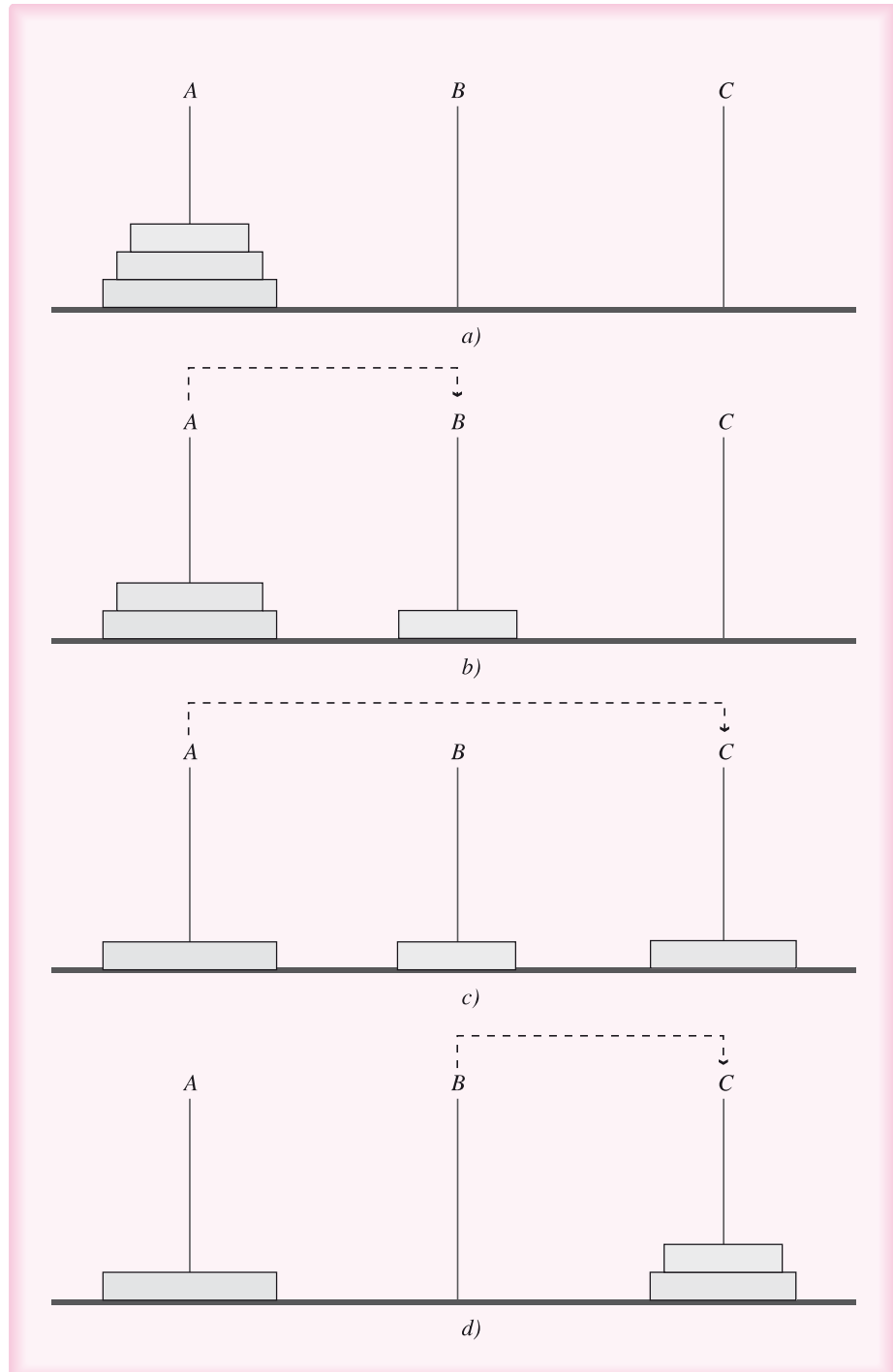
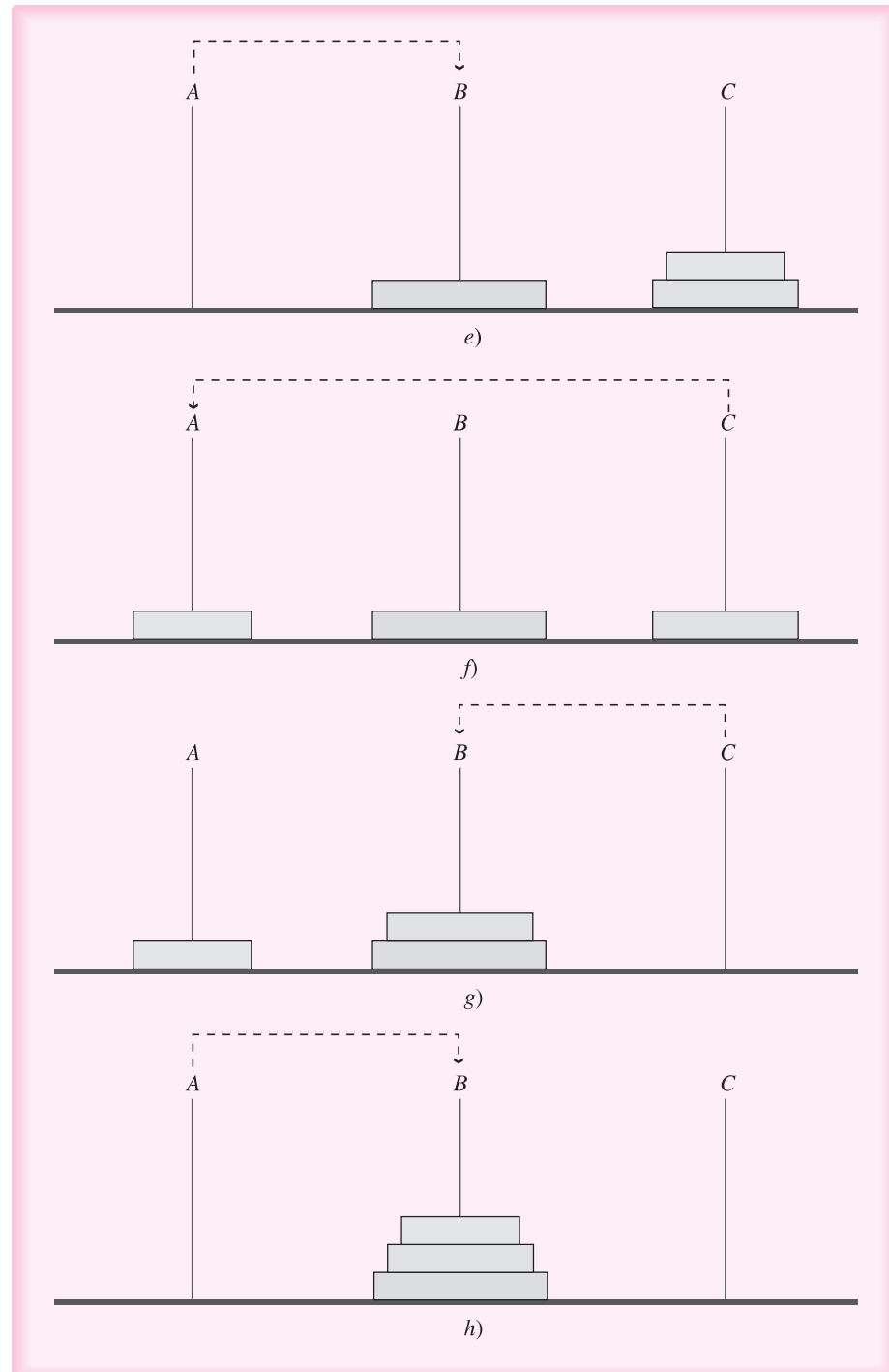


FIGURA 4.15

(continuación)
 e) Luego de mover un disco de A a B.
 f) Luego de mover un disco de C a A.
 g) Luego de mover un disco de C a B.
 h) Estado final.



Algoritmo 4.13 Hanoi

Hanoi (N , ORIGEN, DESTINO, AUXILIAR)

{Este algoritmo obtiene y escribe los movimientos que se deben realizar para transferir los N discos de la torre ORIGEN a la torre DESTINO, con ayuda de la torre AUXILIAR}

1. Si ($N = 1$)

entonces

Escribir "Mover un disco de ORIGEN a DESTINO" {Estado básico}

si no

{Mover $N - 1$ discos de la torre ORIGEN a la torre AUXILIAR}

Hanoi ($N - 1$, ORIGEN, AUXILIAR, DESTINO)

Escribir "Mover un disco de ORIGEN a DESTINO"

{Mover $N - 1$ discos de la torre AUXILIAR a la torre DESTINO}

Hanoi ($N - 1$, AUXILIAR, DESTINO, ORIGEN)

2. {Fin del condicional del paso 1}

En la figura 4.16 se muestra el seguimiento del algoritmo para cuatro discos.

FIGURA 4.16

Funcionamiento interno de la recursión: torres de Hanoi.

Hanoi (N , A, B, C)				
4	A	B	C	
3	A	C	B	
2	A	B	C	
1	A	C	B	[M1]: A → C
[1]	1	C	B	A [M3]: C → B
[2]	2	B	C	A
1	B	A	C	[M5]: B → A
[3]	1	A	C	B [M7]: A → C
[4]	3	C	B	A
2	C	A	B	
1	C	B	A	[M9]: C → B
[5]	1	B	A	C [M11]: B → A
[6]	2	A	B	C
1	A	C	B	[M13]: A → C
[7]	1	C	B	A [M15]: C → B

Movimientos			
[M1] A → C	[M9] C → B		
[M2] A → B	[M10] C → A		
[M3] C → B	[M11] B → A		
[M4] A → C	[M12] C → B		
[M5] B → A	[M13] A → C		
[M6] B → C	[M14] A → B		
[M7] A → C	[M15] C → B		
[M8] A → B			

	Pila	
	Escribir A → B	M [14]
[7]	1 C B A	
	Escribir C → A	M [10]
[5]	1 B A C	
	Escribir C → B	M [12]
[6]	2 A B C	
	Escribir B → C	M [6]
[3]	1 A C B	
	Escribir A → B	M [2]
[1]	1 C B A	
	Escribir A → C	M [4]
[2]	2 B C A	
	Escribir A → B	M [8]
[4]	3 C B A	

El algoritmo recursivo anterior ofrece una solución clara y compacta al problema de las Torres de Hanoi. Sin embargo, es posible también presentar una solución iterativa a este problema, pero es conveniente mencionar que en este caso es más complicada y extensa.

Un subprograma generalmente trabaja con variables locales y parámetros. Cuando se hace una llamada recursiva al subprograma, los valores actuales de variables y parámetros se deben conservar. Además se debe almacenar la dirección a la cual se tendrá que regresar el control una vez que se termina de ejecutar.

El algoritmo anterior trabaja con cuatro parámetros: N , ORIGEN, DESTINO y AUXILIAR. Para construir el algoritmo iterativo, si se quieren conservar los valores de los parámetros en cada llamada recursiva, se deberá definir una pila para cada uno de ellos. Otra alternativa sería definir una única pila, en la que cada elemento fuera capaz de almacenar los cuatro parámetros. En el algoritmo que se va a presentar se trabaja con cuatro pilas:

PILAN: para almacenar imágenes de N .
 PILAO: para almacenar imágenes de ORIGEN.
 PILAD: para almacenar imágenes de DESTINO.
 PILAX: para almacenar imágenes de AUXILIAR.

También será necesario manejar un TOPE para las pilas. El siguiente algoritmo presenta una solución iterativa al problema de las Torres de Hanoi.

Algoritmo 4.14 Hanoi_ite

Hanoi_ite (N , ORIGEN, DESTINO, AUXILIAR)

{Este algoritmo resuelve el problema de las Torres de Hanoi de manera no recursiva. ORIGEN, DESTINO y AUXILIAR son parámetros que representan las tres torres. N simboliza el número de discos}

{PILAN, PILAO, PILAD y PILAX son estructuras de datos tipo pila. TOPE es una variable de tipo entero. VARAUX es una variable de tipo carácter. BAND es una variable de tipo booleano}

1. Hacer TOPE $\leftarrow 0$ y BAND $\leftarrow$ FALSO
2. Mientras $((N > 0)$ y $(BAND = FALSO))$ Repetir
 - 2.1 Mientras $(N > 1)$ Repetir
 - Hacer TOPE $\leftarrow$ TOPE + 1, PILAN[TOPE] $\leftarrow N$, PILAO[TOPE] $\leftarrow$ ORIGEN,
 PILAD[TOPE] $\leftarrow$ DESTINO, PILAX[TOPE] $\leftarrow$ AUXILIAR
 {Simula llamada a Hanoi con $N - 1$, ORIGEN, AUXILIAR y DESTINO}
 Hacer $N \leftarrow N - 1$, VARAUX $\leftarrow$ DESTINO, DESTINO $\leftarrow$ AUXILIAR
 y AUXILIAR $\leftarrow$ VARAUX
 - 2.2 {Fin del ciclo del paso 2.1}
 Escribir "Mover un disco de ORIGEN a DESTINO"
 Hacer BAND $\leftarrow$ VERDADERO
 - 2.3 Si $(TOPE > 0)$ {Las pilas no están vacías}

entonces

```
{ Se extraen los elementos del TOPE de la pila }
Hacer  $N \leftarrow \text{PILAN}[\text{TOPE}]$ ,  $\text{ORIGEN} \leftarrow \text{PILAO}[\text{TOPE}]$ ,
     $\text{DESTINO} \leftarrow \text{PILAD}[\text{TOPE}]$ ,  $\text{AUXILIAR} \leftarrow \text{PILAA}[\text{TOPE}]$ 
    y  $\text{TOPE} \leftarrow \text{TOPE} - 1$ 
Escribir "Mover un disco de ORIGEN a DESTINO"
{ Simula llamada a Hanoi con  $N - 1$ , AUXILIAR, DESTINO y ORIGEN }
Hacer  $N \leftarrow N - 1$ ,  $\text{VARAUX} \leftarrow \text{ORIGEN}$ ,  $\text{ORIGEN} \leftarrow \text{AUXILIAR}$ ,
     $\text{AUXILIAR} \leftarrow \text{VARAUX}$  y  $\text{BAND} \leftarrow \text{FALSO}$ 
```

2.4 {Fin del condicional del paso 2.3}

3. {Fin del ciclo del paso 2}

En la figura 4.17 se muestra el seguimiento del algoritmo para tres discos.

FIGURA 4.17.
Seguimiento del algoritmo
iterativo: Torres de Hanoi.

N	ORIGEN	DESTINO	AUXILIAR	TOPE	BAND	VARAUX
3	A	B	C	0	F	
2		C	B	1		B
1		B	C	2	V	C
2	A	C	B	1		
1	B		A		F	A
3	A	B	C	0	V	
2	C		A		F	A
1		A	B	1	V	B
2	C	B	A	0		
1	A		C		F	C
					V	

Nota: en el lugar de VERDADERO se escribe V,
en el lugar de FALSO F.

Movimientos					Pila N	Pila O	Pila D	Pila A
$A \rightarrow B$								
$A \rightarrow C$								
$B \rightarrow C$								
$A \rightarrow B$								
$C \rightarrow A$								
$C \rightarrow B$								
$A \rightarrow B$								
$A \rightarrow C$								

Topo $\rightarrow$ 1	2	C	B	A				
Topo $\rightarrow$ 2	2	A	C	B				
Topo $\rightarrow$ 1	3	A	B	C				