Práctica Número 2
Diseñar recursivamente y transformar a iterativo

1. Introducción

En 1883 empezó a venderse en Francia un antiguo rompecabezas oriental, rescatado para Occidente por el profesor N. Claus (de Siam) y cuyas primeras referencias eran los escritos del ilustre mandarín Fer-Fer-Tam-Tam. Según una leyenda india, en el Templo de Benarés, bajo el domo que marca el centro del mundo, hay una placa de latón con tres agujas de diamante. Durante la creación, Dios puso sesenta y cuatro discos de oro puro de distinto tamaño en una de las agujas, formando una torre. Los bramanes llevan generaciones cambiando de lugar, uno a uno, los discos de la torre entre las tres agujas de forma que en ningún momento un disco mayor descanse sobre otro más pequeño. Cuando hayan conseguido trasladar todos los discos a otra aguja su trabajo estará terminado, y la torre y el templo se derrumbarán, y con un gran trueno, el mundo se desvanecerá. La versión simplificada que se vendía en Francia se componía de ocho discos de madera.

En realidad, la Torre de Hanoi y la leyenda india habían sido inventadas por el matemático francés Édouard Lucas (N. Claus de Siam es un anagrama de Lucas d'Amiens). Su compatriota, el escritor Henri de Parville amplió y adornó la leyenda poco tiempo después. A pesar de que el reto planteado es relativamente sencillo, la idea de Lucas ha demostrado ser una de las más fecundas de la historia de las matemáticas recreativas.

Si no lo has hecho antes, antes de seguir leyendo tal vez deberías familiarizarte con el rompecabezas y resolverlo por ti mismo. Puedes usar un modelo real o uno de los muchos simuladores que hay disponibles en Internet.

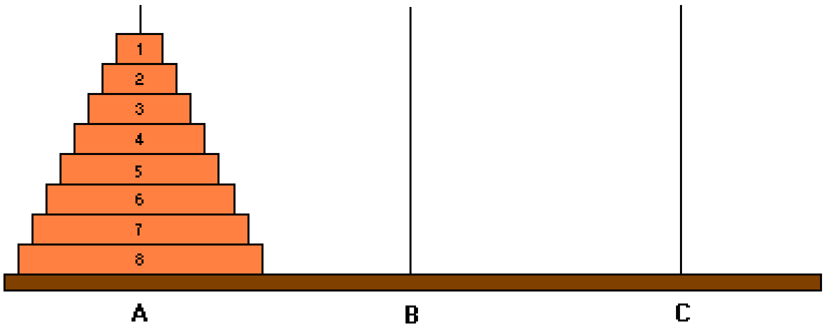
2. Notación

Los discos se numerarán de 1 a 8 (o a n, en general), empezando por el más pequeño. Los postes (que se supondrán alineados de izquierda a derecha) serán marcados con letras mayúsculas (A, B y C). El inicial será A y el objetivo C.

3. Un algoritmo recursivo

La Torre de Hanoi suele aparecer como ejemplo para ilustrar el concepto de recursión en los cursos de programación, ya que existe un algoritmo recursivo sencillo de descubrir que lo resuelve. Supongamos que queremos trasladar los ocho discos del poste A al poste C. Como el disco 8 siempre está abajo del todo, la única forma de hacerlo es trasladar primero la torre de siete discos 1...7 al poste B. Entonces podremos llevar el disco 8 de A a C, y para terminar tendremos que trasladar de nuevo la torre 1...7, ahora de B a C.

Pero los discos 1...7 forman una torre totalmente similar a la inicial, así que en dos de los tres pasos anteriores nos enfrentamos con un problema análogo al original. De hecho,



puede resolverse de la misma forma, trasladando ahora la torre 1...6. Por ejemplo, el primer paso (trasladar la torre 1...7 de A a B) se puede descomponer en estos tres pasos.

Por supuesto, en dos de estos tres pasos nos volvemos a encontrar con el problema original, ahora con $n = 6$. El proceso no es infinito, ya que llega el momento en que trasladar una torre equivale a trasladar un solo disco (esto ocurre cuando la torre es de un solo disco). En resumen, el algoritmo recursivo es el siguiente.

Algoritmo para trasladar la torre $1 \dots n$ del poste X al poste Z, usando como auxiliar el poste Y.

1. Si $n = 1$, lleva el disco 1 de X a Z y termina.
2. Traslada la torre $1 \dots n-1$ usando este mismo algoritmo, de X a Y, usando como auxiliar Z.
3. Lleva el disco n de X a Z.
4. Traslada la torre $1 \dots n-1$ usando este mismo algoritmo, de Y a Z, usando como auxiliar X.

Ésta es más la definición de un problema que su resolución. En realidad, lo único que hace es especificar unos pasos inevitables. Por ejemplo, como vimos antes, para resolver el puzzle es obligatorio llevar el disco n de A a C, y para ello es obligatorio llevar antes la torre $1 \dots n-1$ a B, etc. La secuencia de movimientos que produce este algoritmo es la única solución óptima (o sea, de longitud mínima) posible. El número de movimientos es

$$M(n) = 2^n - 1,$$

como se puede demostrar fácilmente por inducción sobre el número de discos.

Demostración.
Se cumple para $n = 1$ pues

$$M(1) = 1 = 2^1 - 1.$$

Supongamos cierto para n, y veamos que se cumple para $n + 1$.
Al ejecutarse el algoritmo para $n + 1$ se llama a sí mismo dos veces para n, más un movimiento del disco $n + 1$. Así que:

$$M(n+1) = 2M(n) + 1 = 2(2^n - 1) + 1 = 2^{n+1} - 2 + 1 = 2^{n+1} - 1.$$

$\square$

Para $n = 8$ el número de movimientos es de $2^8 - 1 = 255$. Para $n = 64$, de $264 - 1 =$ 18,446,744,073,709,551,615. Si los bramanes de Benarés cambiaran un disco de sitio cada segundo necesitarían más de quinientos ochenta mil millones de años para terminar su tarea, unas cuarenta veces la edad del Universo.

4. Actividad a realizar

1. Implementar una versión recursiva que lea por teclado el número de discos y muestre por pantalla los movimientos a realizar

2. Realizar un análisis experimental de la eficiencia y constatar que coincide con el análisis teórico