Leonardo Abril Bejarano
Servicios y procesos

Introduccion

El matemático francés Édouard Lucas d'Amiens con el pseudónimo de profesor N. Claus de Siam (anagrama de Lucas d'Amiens), mandarín del Colegio de Li–Sou-Stian (una nueva permutación de letras, esta vez de las de las de las palabras Saint Louis), ideó y dio a conocer este problema en 1883. Se comercializó como un juego con el nombre de "La Torre de Hanói".

El material del rompecabezas lo forman tres pivotessujetos en una base horizontaly un cierto número de discos de distintos diámetros que se colocan en uno de los pivotes extremos. En la parte baja se coloca el de mayor diámetro y encima los de diámetros menores en orden decreciente. El objetivo del juego consiste en pasar los discos de un extremo al otro siguiendo unas precisas normas que son:

•En cada movimiento solo puede moverse un disco.
•El número de movimientos debe ser el menor posible
•No se puede colocar nunca un disco sobre otro de me
nor diámetro.

Teoria de los Grafos

Un juego TH_5, como ya sabemos, requiere de $M_5=2^5-1=31$ movimientos para ser resuelto de manera óptima, es decir, con el menor número de movimientos. Cada movimiento, como también se ha dicho ya, puede ser representado mediante un código en base 3 con cinco dígitos. En cambio, con cinco dígitos existen $3^5=243$ códigos diferentes.
¿Representan algo los códigos no usados en el camino óptimo?
Cada uno de los códigos no usados representa una situación del juego por la que no se pasa en el camino óptimo, pero que eventualmente podría darse si elegimos un camino más largo. En realidad, como vamos a ver, se puede elegir un camino que recorra todas las situaciones posibles del juego y, por tanto, todos los códigos posibles. Este camino, que llamaremos pésimo, como hace Rodolfo Valeiras, tendrá 3^n-1 movimientos. La existencia de este camino nos proporciona el resultado siguiente: sea cual sea la posición del juego (siempre que cumpla sus normas) podremos llegar, mediante movimientos legales, a la solución. Para su análisis nos ayudaremos del llamado Grafo de Hanoi, invención de Ian Stewart.

Empezaremos considerando un juego TH_1, es decir, un juego donde sólo hay un disco. Tenemos que $x_0=0$ (en notación de base 3). Del poste 0 podemos pasar al 1 ò al 2, aunque ya sabemos que lo más eficiente es pasar al 2. Si optamos por pasar al poste 1, desde éste se puede pasar al poste 2, completando el camino pésimo en dos movimientos. Estas opciones se pueden representar en forma de triángulo, como muestra la figura 5. El lado derecho de este triángulo (que une 0 con 2) es el camino óptimo. El camino pésimo consiste en recorrer los vértices 0-1-2.

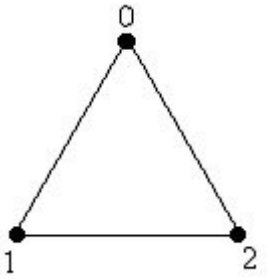

Figura 5: Grafo de Hanoi para el juego TH_1

Supongamos ahora que el juego tiene 2 discos. Entonces, de la posición 00 se puede pasar a 01 ó a 02 (sólo se puede mover el pequeño). De 01 podemos a 02, cerrando así un triángulo. Pero también de 01 se puede pasar a 21 y de 02 se puede pasar a 12. Ahora se analizan los movimientos legales desde 21 y desde 12 y el resultado puede presentarse en un grafo como el de la figura 6.

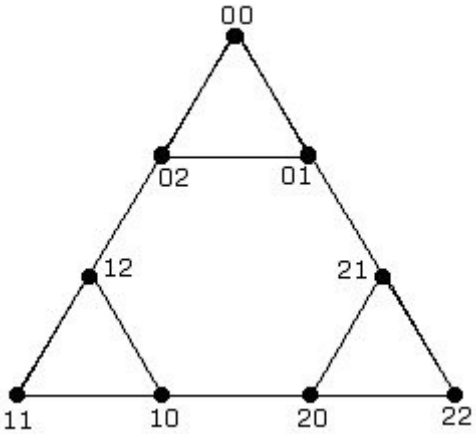

Figura 6: Grafo de Hanoi para el juego TH_2

Como se puede observar en esta figura, todas las posibles posiciones del juego están conectadas. El lado derecho del triángulo nos sigue presentando la solución óptima (en tres movimientos, por supuesto). Para encontrar el camino pésimo debemos recorrer todo el grafo sin pasar dos veces por el mismo vértice. Esto es siempre posible. Para ello, saliendo desde 00, recorremos el primer triángulo en sentido horario, terminando en 02. Luego bajamos al vértice 12 y recorremos ese triángulo en sentido antihorario, terminando en 10. Llegamos luego al vértice 20 y recorremos ese triángulo en sentido horario. Este recorrido, dicho sea para que Euler repose tranquilo, como se ve, pasa por todos los puntos, pero no por todas las aristas, es decir, el juego pasa por todas las posiciones posibles pero no se realizan todos los movimientos posibles. Si quisiéramos, además, realizar todos los movimientos posibles una sola vez, habría que recorrer el grafo pasando por todas las aristas una sola vez, cosa que es imposible pues existen más de dos vértices de índice impar lo que imposibilita, como Euler nos enseñó, tal recorrido. Agotar todos los movimientos exigiría repetir alguno de ellos. Tal camino merecería ser llamado el peor camino.

El grafo para el juego TH_2 puede ser entendido como un gran triángulo equilátero que tiene en cada vértice sendos triángulos equiláteros colocados en su interior. Los lados de los triángulos pequeños representan los movimientos del disco pequeño y los tramos no usados de los lados del grande representan los movimientos del disco grande. Es decir, el grafo de TH_2 consiste en un triángulo que tiene en cada vértice un grafo de TH_1. Esta forma recursiva de mirar el grafo nos permitirá construir los siguientes. En efecto, el grafo de TH_3 (figura 7) es un triángulo que tiene en cada vértice un grafo de TH_2.

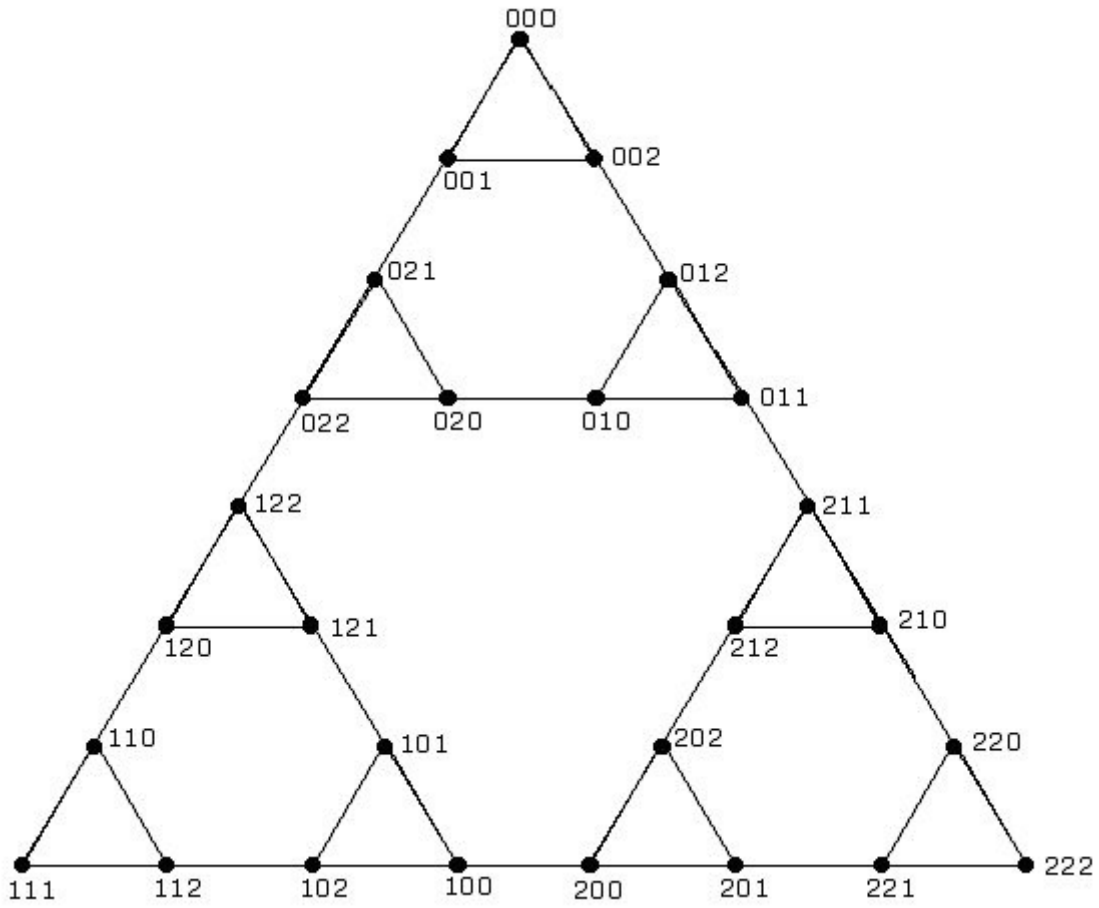

Figura 7: Grafo de Hanoi para el juego TH_3

La forma de encontrar el camino pésimo en este grafo es también recurrente y se puede encontrar de manera sencilla si recorremos las siguientes etapas:

i) Si n es par recorremos el primer triángulo pequeño en sentido horario. Si n es impar lo recorremos en sentido antihorario.

ii) Pasamos al único triángulo pequeño accesible desde el fin del recorrido anterior y lo recorremos en el sentido contrario al anterior.

iii) Repetimos el paso ii tantas veces como sea necesario hasta llegar a la última posición.

Cada lado de un triángulo pequeño representa un movimiento del disco pequeño, cada lado de un triángulo mediano que no pertenezca a un triángulo pequeño representa un movimiento del disco segundo y cada lado del triángulo grande que no pertenezca a un triángulo menor representa los movimientos del disco grande.

¿Se atreverá el lector a diseñar un grafo de Hanoi para TH_4? Nosotros damos en la figura 8 el grafo de TH_5 sin indicar los códigos. Un buen ejercicio sería, precisamente, ponérselos.

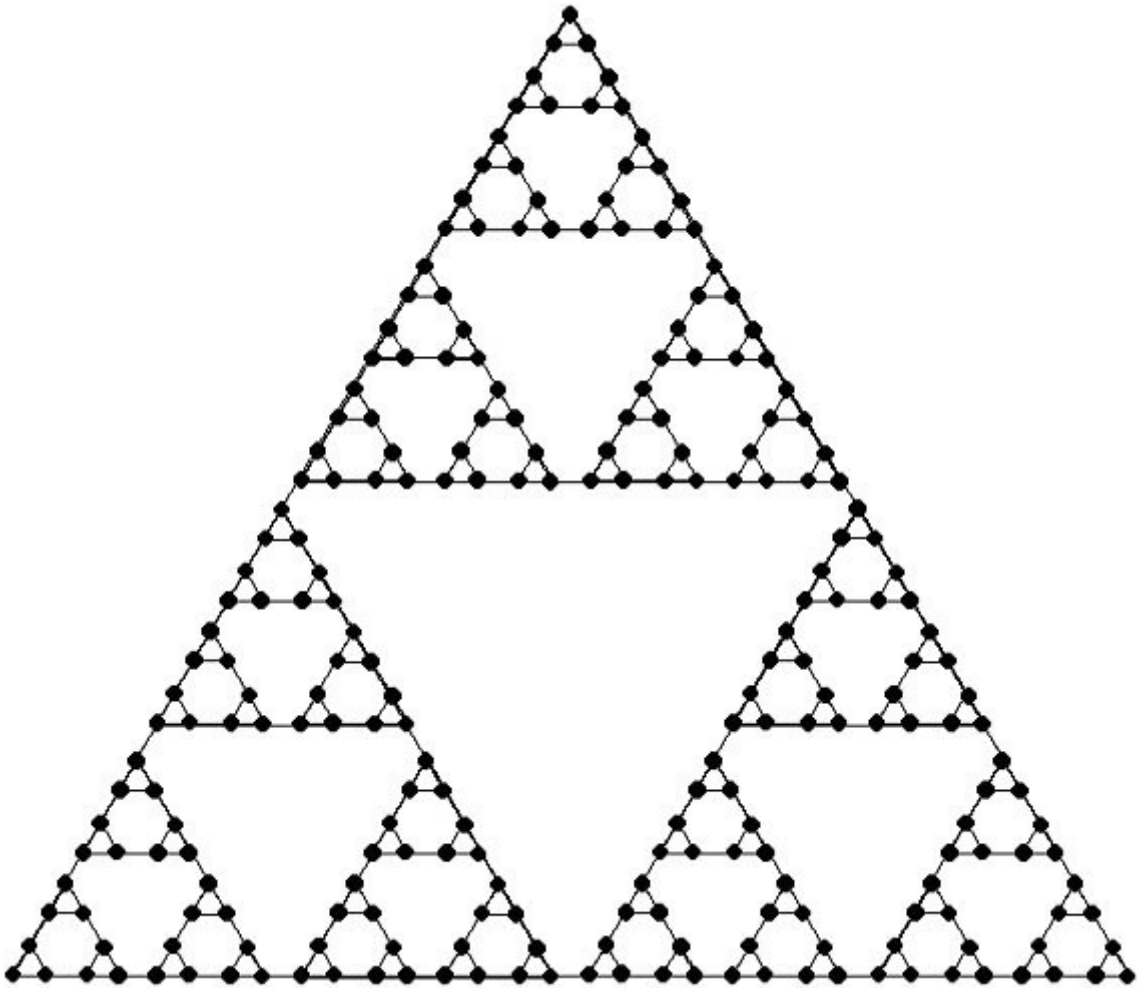

Figura 8: Grafo de Hanoi para el juego TH_5

Por supuesto que el grafo de Hanoi para el juego TH_n, cuando n vaya creciendo sin límite, se convierte en un fractal. Cada una de sus partes reproduce el modelo del grafo completo.

Vamos a mencionar una curiosa relación. Si construimos un triángulo de Pascal ilimitado y sustituimos cada número par por un punto blanco y cada número impar por un punto negro, el resultado es un fractal bastante parecido al famoso estuche de Sierpinski. Para crear un estuche de Sierpinski dibuje un triángulo negro. Divídalo en cuatro triángulos iguales y pinte de blanco el central. Haga lo mismo con los tres triángulos negros pequeños y repita este proceso hasta el infinito. El resultado es bastante parecido a colorear de la forma indicada el triángulo de Pascal. El hecho es que Édouard Lucas, el creador de la Torre de Hanoi, estudió la forma de averiguar cuándo es par o impar un número del triángulo de Pascal, de forma que también contribuyó, sin saberlo, a la creación de estos fractales. Nos interesa ahora, a modo de despedida, que el lector reflexione sobre la siguiente pregunta: ¿No se parece el estuche de Sierpinski (figura) al grafo de Hanoi?

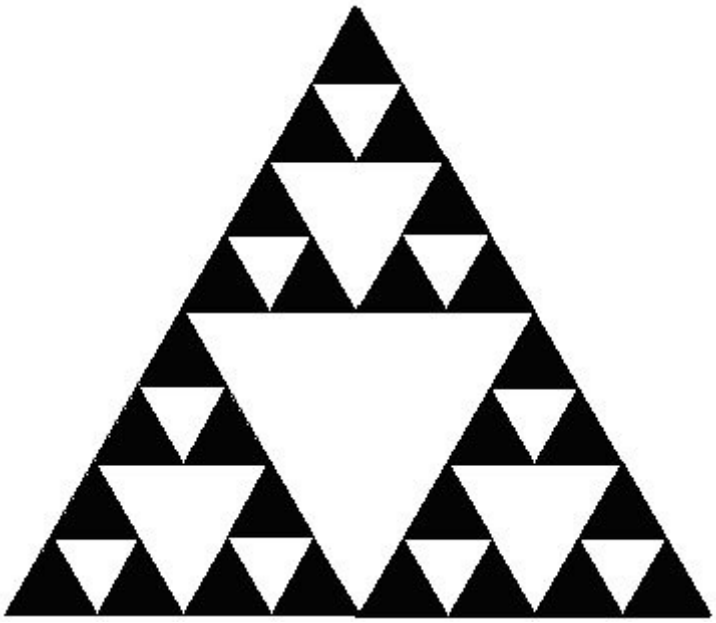

Figura 9: Estuche de Sierpinski

Diseño del programa

Explicacion del juego.

El planteamiento de este programa consiste en representar el grafo de las torres de hanoi en un conjunto dde clases.

Las posiciones se representan como mapas. y cada mapa tendra n submapas.

El limite sera el numero de movimientos que no podra superar el maximo de movimientos y que se calcula con 2^n-1.

Los submapas de diferente nivel se mezclaran en una misma pila y cuando un submapa o conjunto de submapas mueran,

hara backtracking y seguira con un submapa de un nivel inferior.

La aplicacion utiliza backtracking, con lo que ahora se utilizara los conceptos de mapa padre y mapa hijo

para poder realizarlo.

La aplicacion debe reciclar todo nodo que sea repetitivo de algun nodo padre, ademas de utilizar backtracking a

la inversa y probar de ver en que niveles de backtracking es mas optimo. En general, esta actualizacion busca optimi-

zar al maximo la aplicacion con estas dos nuevas caracteristicas.

El planteamiento anterior sin backtracking inverso era compararlo con un unico mapa resultado. Con lo cual

solo necesitaba un motor de juego y no para hasta encontrar el resultado. Ahora, al empezar desde dos puntos de partida

necesitaremos dos motores para poder realizar el backtracking inverso. El motor inverso se encargara de realizar todos los

caminos posible mediante un parametro limite, y guardaremos en una arraylist los resultados. El motor normal

sera el mismo, pero ahora comparara el resultado con un arraylist de las pilas que hayamos sacado del motor inverso(solo

el mapa con mas nivel)

El esquema UML del programa es el siguiente: