

## Torres de Hanoi

Para el programa ensamblador hemos decidido utilizar el algoritmo de las torres de Hanoi. Por ello, para adentrarnos en este tema conoceremos un poco de su historia.

Las torres de Hanoi es un rompecabezas matemático inventado por Eduard Lucas en 1883, creó el juego y lo bautizó en honor a la ciudad vietnamita de Hanoi. Sin embargo, el juego también tiene raíces en la mitología hindú, donde se dice que los dioses utilizaron una torre similar para transferir discos de oro de un poste a otro.

En su forma básica, las Torres de Hanoi consisten en tres postes verticales y un conjunto de discos de diferentes tamaños, que se colocan inicialmente en el poste izquierdo en orden decreciente de tamaño. El objetivo es mover todos los discos al poste derecho, utilizando el poste central como un espacio de trabajo temporal.

Las instrucciones del juego son las siguientes:

1. Solo se puede mover un disco a la vez.
2. Un disco de mayor tamaño no puede estar sobre un disco más pequeño que el mismo.

Además, el jugador debe tratar de minimizar el número de movimientos necesarios para completar el rompecabezas.

Dependiendo del número de discos es la dificultad del juego. El número de movimientos requeridos para resolver el rompecabezas con  $n$  discos se puede calcular mediante la fórmula  $2^n - 1$ , lo que significa que el número de movimientos aumenta exponencialmente con el número de discos. En esta ocasión, utilizaremos este algoritmo para 3 discos, por lo que:

$$2^3 - 1 = 7$$

En este sentido, tendríamos 7 movimientos para resolver este problema con 3 discos. Es decir, la cantidad de movimientos se tiene que optimizar para no malgastar ningún movimiento y así, poder ganar el juego.

La solución del rompecabezas de las Torres de Hanoi de tres discos no es difícil, pero aún así requiere una lógica y un razonamiento cuidadosos. Para resolverlo, el jugador debe seguir una serie de pasos cuidadosamente diseñados.

En primer lugar, el jugador debe mover el disco más pequeño del poste izquierdo al poste derecho. Luego, debe mover el segundo disco, es decir, el mediano al poste central. Una vez que se ha hecho esto, el jugador puede mover el disco más pequeño del poste derecho al poste central.

Luego, el jugador debe mover el disco más grande del poste izquierdo al poste derecho. Después de hacer esto, el jugador puede mover el disco más pequeño del poste central al poste izquierdo. A continuación, se puede mover el disco mediano del poste central al poste derecho.

Finalmente, el jugador puede mover el disco más pequeño del poste izquierdo al poste derecho, y el juego estará completo.

Las Torres de Hanoi al ser un juego de estrategia matemático, se puede aplicar en la programación, y específicamente en el lenguaje de programación ensamblador (ASM). En ASM, el juego se puede implementar como un algoritmo recursivo que utiliza el stack para almacenar la información sobre los movimientos de los discos.

En lenguaje ensamblador, el algoritmo de las Torres de Hanoi de tres discos consta de tres funciones: una función principal que llama a una función recursiva para mover los discos, una función recursiva que mueve los discos y una función auxiliar que imprime los movimientos en la pantalla. La función recursiva utiliza el stack para almacenar información sobre los movimientos y llama a sí misma para mover los discos más pequeños.

Para implementar el algoritmo en ASM, se pueden utilizar instrucciones como PUSH, POP, CALL y RET para manipular el stack y la pila de llamadas. La función principal llama a la función recursiva, pasándole los parámetros necesarios para mover los discos. La función recursiva utiliza la pila para almacenar información sobre los movimientos, llamando a sí misma para mover los discos más pequeños hasta que se haya completado el rompecabezas.

En conclusión, el problema de las Torres de Hanoi es un ejemplo clásico de un problema recursivo, lo que significa que se puede resolver mediante la repetición de un conjunto de reglas simples. Así mismo, su implementación en lenguaje ensamblador es importante ya que, ayuda a comprender la programación de bajo nivel.