



- Anthony Perez  
Estructura de Datos

## Actividad 2. Gestión de la práctica y experimentación

### Recursividad y problema de las torres de Hanoi

Revisión Rápida de Conceptos Básicos:

📖 📖 Informe: Torres de Hanoi - Solución Recursiva 📖 📖

#### 1. Descripción del problema

Las Torres de Hanoi es un clásico problema matemático y juego de puzle inventado por el matemático francés Édouard Lucas en 1883. El juego consiste en tres varillas y un número de discos de diferentes tamaños que pueden deslizarse en cualquier varilla.

Reglas:

1. Solo se puede mover un disco a la vez.
2. Cada movimiento consiste en tomar el disco superior de una de las varillas y deslizarlo en otra varilla, sobre los otros discos que puedan estar ya presentes en esa varilla.
3. Ningún disco puede ser colocado encima de un disco más pequeño.

🎯 🎯 Objetivo 🎯 🎯

Mover toda la pila de discos de la varilla de origen a la varilla de destino, utilizando la varilla auxiliar como ayuda, respetando todas las reglas mencionadas.

#### 2. Implementación recursiva (matemática)

La solución recursiva para las Torres de Hanoi se basa en la observación de que para mover  $n$  discos de una varilla a otra, podemos:

1. Mover  $n-1$  discos de la varilla de origen a la varilla auxiliar.
2. Mover el disco más grande de la varilla de origen a la varilla de destino.
3. Mover los  $n-1$  discos de la varilla auxiliar a la varilla de destino.

Matemáticamente, podemos expresar esto como una función recursiva:

Donde  $n$  es el número de discos, y origen, destino y auxiliar son las tres varillas.

### 3. Implementación recursiva (Python)

#### Estructura del algoritmo:

El algoritmo sigue la estructura recursiva descrita en la implementación matemática. Se divide el problema en subproblemas más pequeños, resolviendo el caso base ( $n=1$ ) directamente y llamando recursivamente a la función para los casos más complejos.

Condición base y llamada recursiva:

Condición base: Cuando  $n = 1$ , simplemente se mueve el disco de la varilla de origen a la de destino.

Llamada recursiva: Para  $n > 1$ , se realizan dos llamadas recursivas:

1. Mover  $n-1$  discos de origen a auxiliar.
2. Mover  $n-1$  discos de auxiliar a destino.

#### Complejidad temporal:

La complejidad temporal del algoritmo es  $O(2^n)$ , donde  $n$  es el número de discos. Esto se debe a que para cada  $n$ , se realizan 2 llamadas recursivas con  $n-1$  discos, más una operación constante (mover el disco  $n$ ).

La relación de recurrencia es:

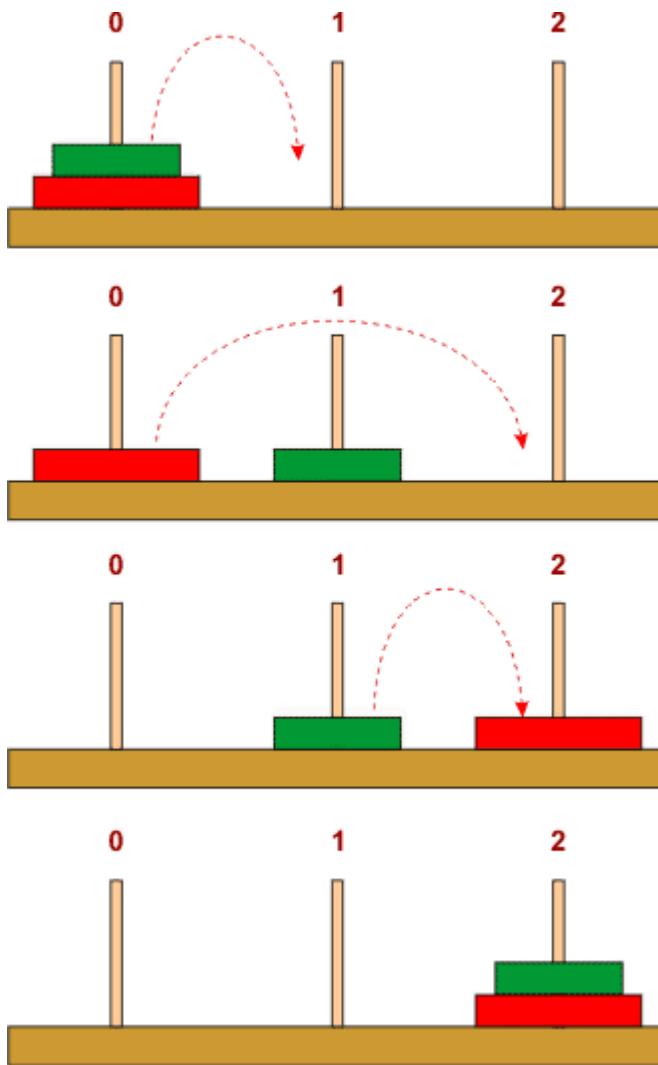
$$T(n) = 2T(n-1) + 1, \text{ con } T(1) = 1$$

Resolviendo esta relación, obtenemos:

$$T(n) = 2^n - 1$$

Por lo tanto, el número de movimientos necesarios para resolver el problema con  $n$  discos es  $2^n - 1$ .

## 5. Documentación



main.py U × README.md M

main.py > ...

```

1  # Problema De Las torres de Hanoi
2
   tabnine: test | explain | document | ask | Codiumate: Options | Test this function
3  def TowerOfHanoi(n , source, destination, auxiliary):
4      if n==1:
5          print ("Mover el disco 1 desde la fuente",source,"A destino",destination)
6          return
7      TowerOfHanoi(n-1, source, auxiliary, destination)
8      print ("Mover Disco",n,"de la fuente",source,"A destino",destination)
9      TowerOfHanoi(n-1, auxiliary, destination, source)
10
11
12  n = 4
13  TowerOfHanoi(n,'A','B','C')
14  # A, C, B are the name of rods

```

## Conclusiones y reflexiones sobre el uso de la recursividad:

- 1. Elegancia y simplicidad:** La solución recursiva para las Torres de Hanoi demuestra cómo problemas aparentemente complejos pueden resolverse con algoritmos sorprendentemente simples y elegantes.
- 2. Poder de la recursividad:** Este problema ilustra perfectamente cómo la recursividad puede descomponer un problema grande en subproblemas más manejables.
- 3. Complejidad vs. Legibilidad:** Aunque la implementación recursiva es concisa y fácil de entender, su complejidad exponencial la hace ineficiente para un gran número de discos. Esto destaca la importancia de considerar tanto la elegancia del código como su eficiencia.
- 4. Pensamiento recursivo:** Resolver este problema ayuda a desarrollar el "pensamiento recursivo", una habilidad valiosa en programación y resolución de problemas en general.
- 5. Aplicaciones prácticas:** Aunque las Torres de Hanoi es un problema teórico, las técnicas utilizadas para resolverlo (dividir problemas en subproblemas, recursividad) tienen aplicaciones prácticas en muchos campos de la informática y las matemáticas.



Para comprender mejor el código puedes revisar aquí

<https://github.com/TONY24/Hanooi.git>

