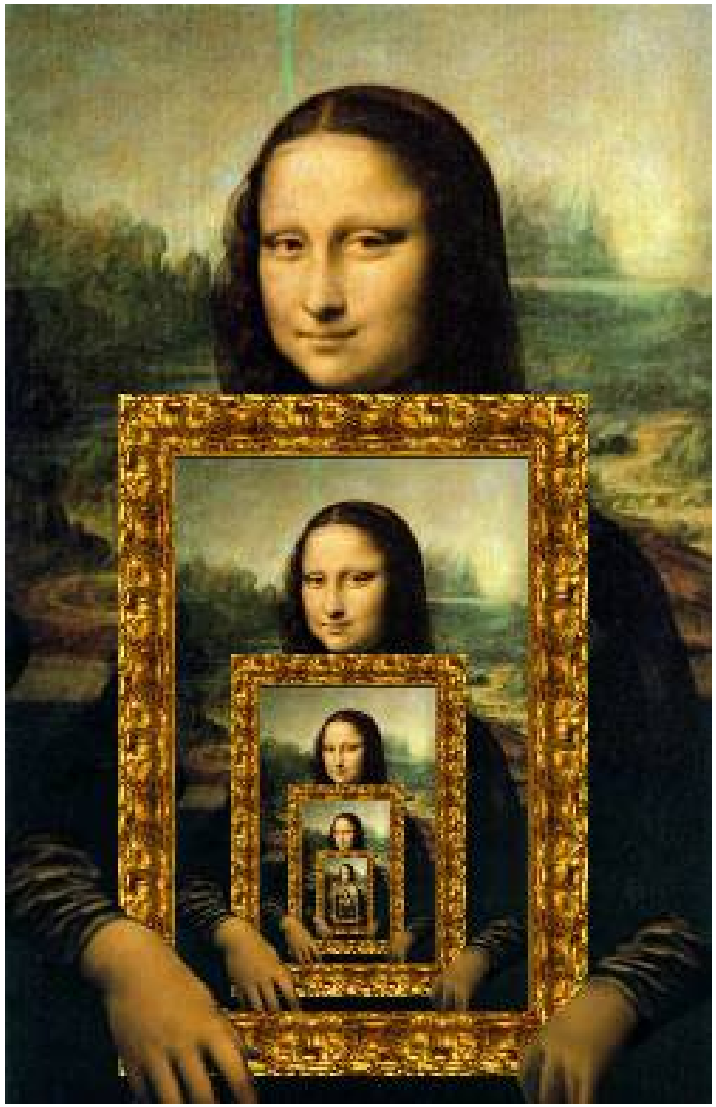
Recursão

Marcelo Hashimoto

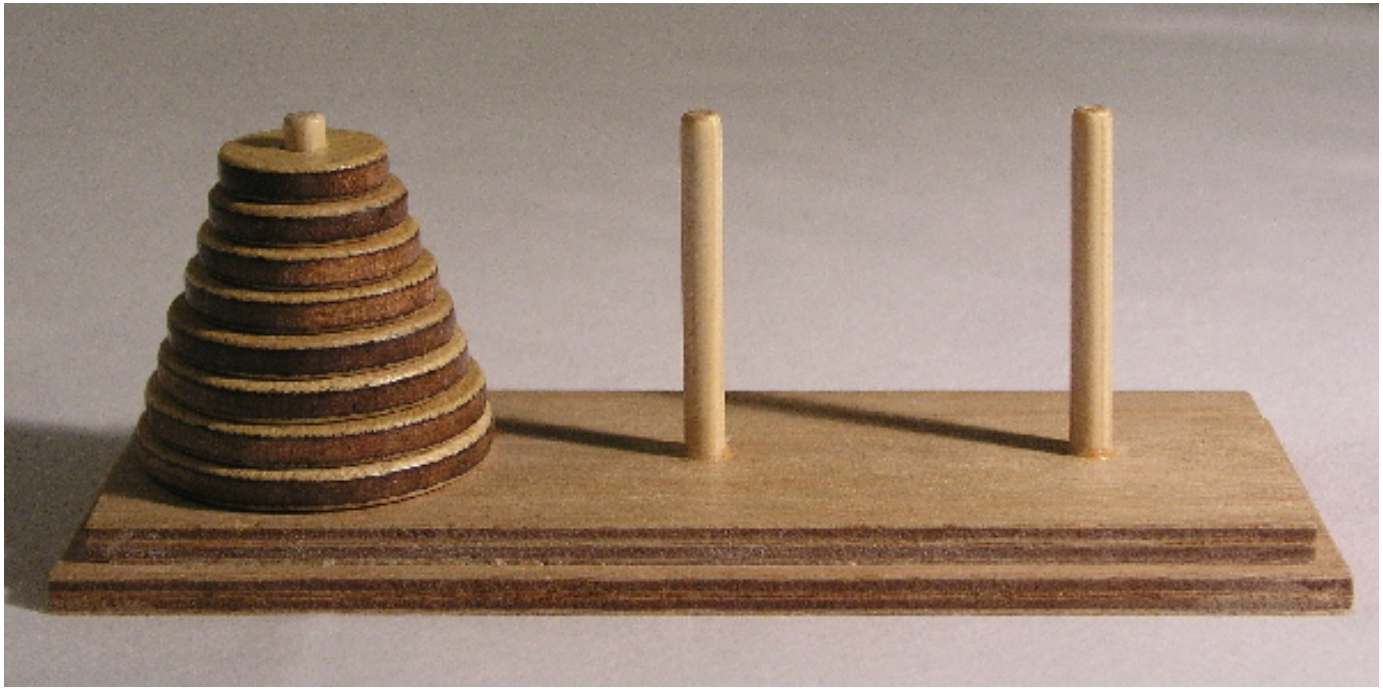
Recursão



Um algoritmo é recursivo quando chama a si próprio.

Um algoritmo é recursivo quando chama a si próprio.
(mais especificamente, versões menores de si próprio)

Torres de Hanói



Torres de Hanói

- 3 torres e 64 discos de tamanhos diferentes.

Torres de Hanói

- 3 torres e 64 discos de tamanhos diferentes.
- Todos os discos começam na primeira torre. (origem)

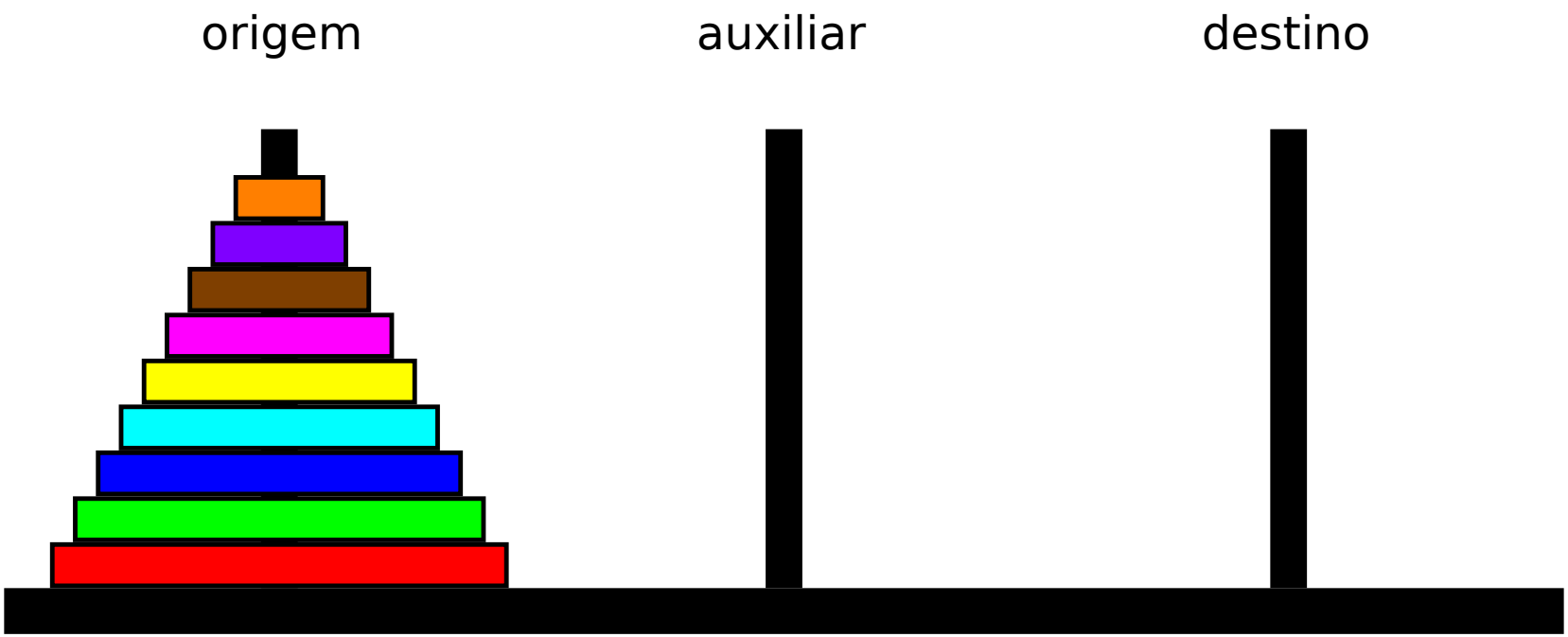
Torres de Hanói

- 3 torres e 64 discos de tamanhos diferentes.
- Todos os discos começam na primeira torre. (origem)
- O objetivo é movê-los para a terceira torre. (destino)

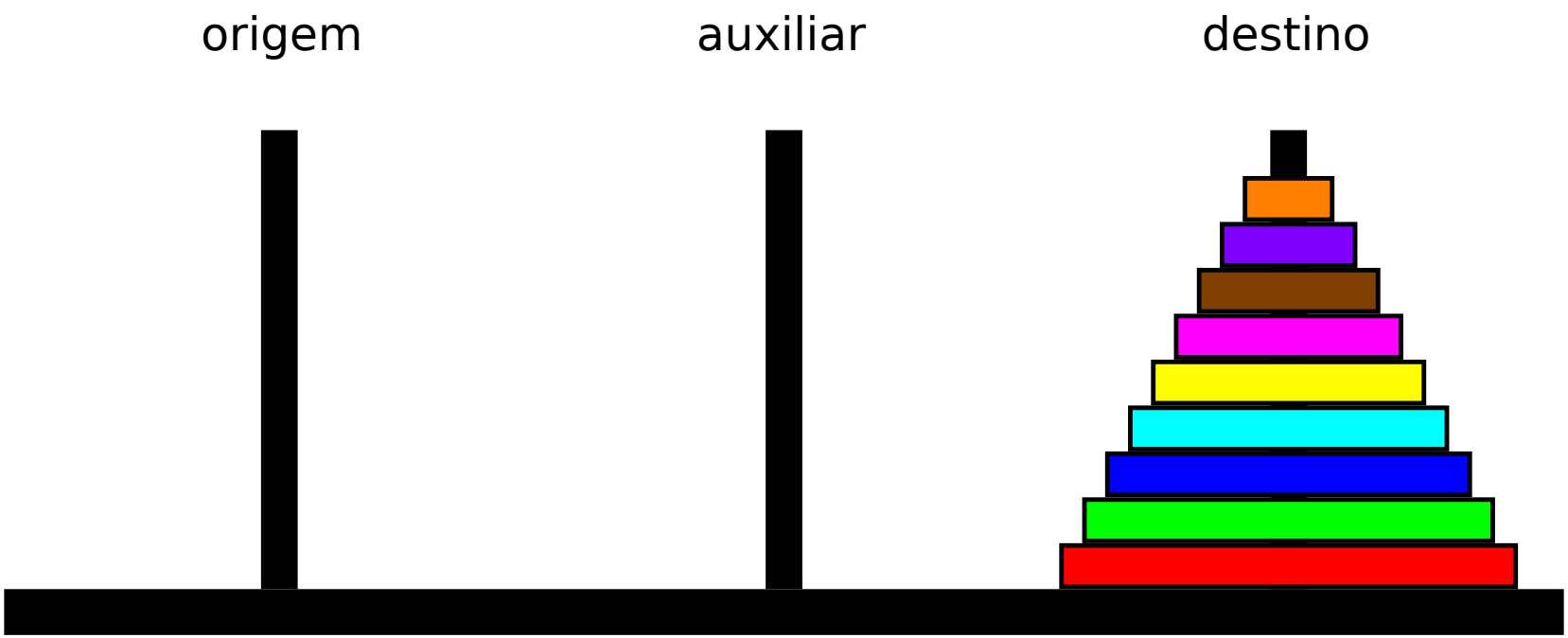
Torres de Hanói

- 3 torres e 64 discos de tamanhos diferentes.
- Todos os discos começam na primeira torre. (origem)
- O objetivo é movê-los para a terceira torre. (destino)
- A segunda torre pode ser utilizada para ajudar. (auxiliar)

Torres de Hanói



Torres de Hanói



Torres de Hanói

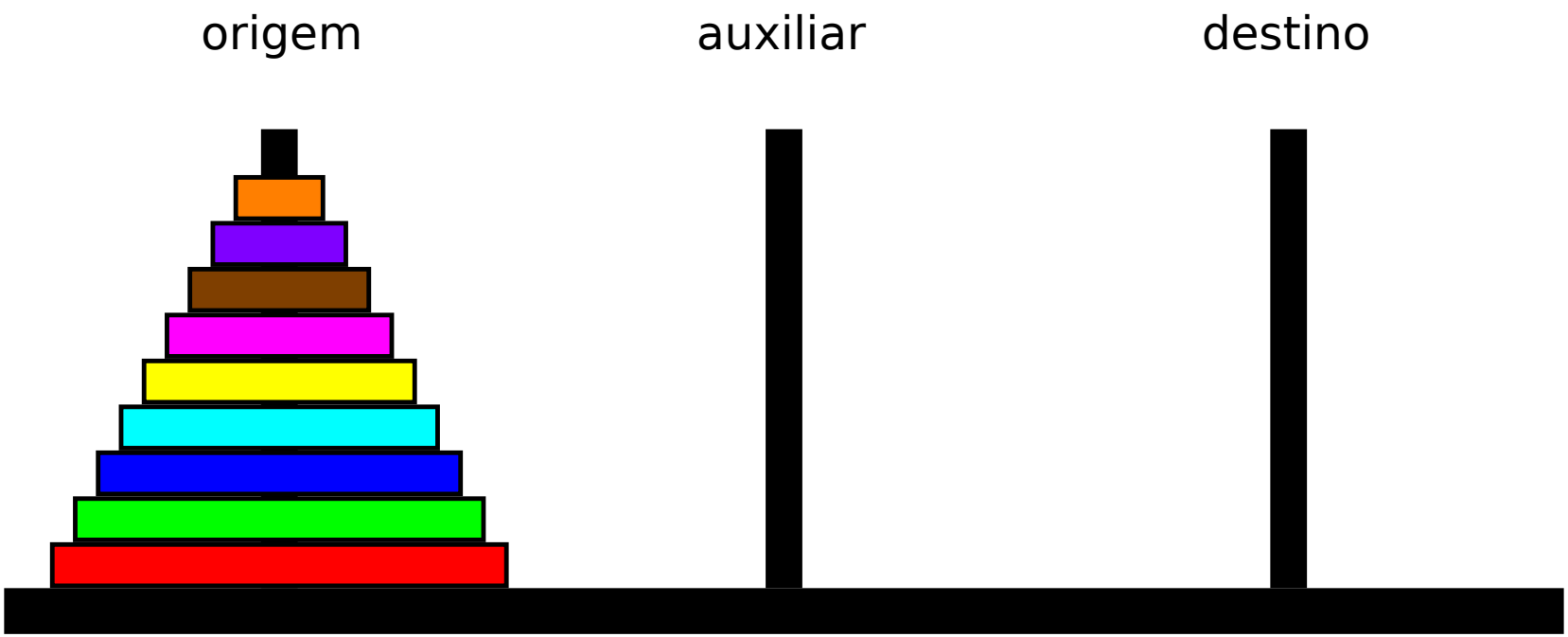
- Apenas um disco pode ser movido de cada vez.

Torres de Hanói

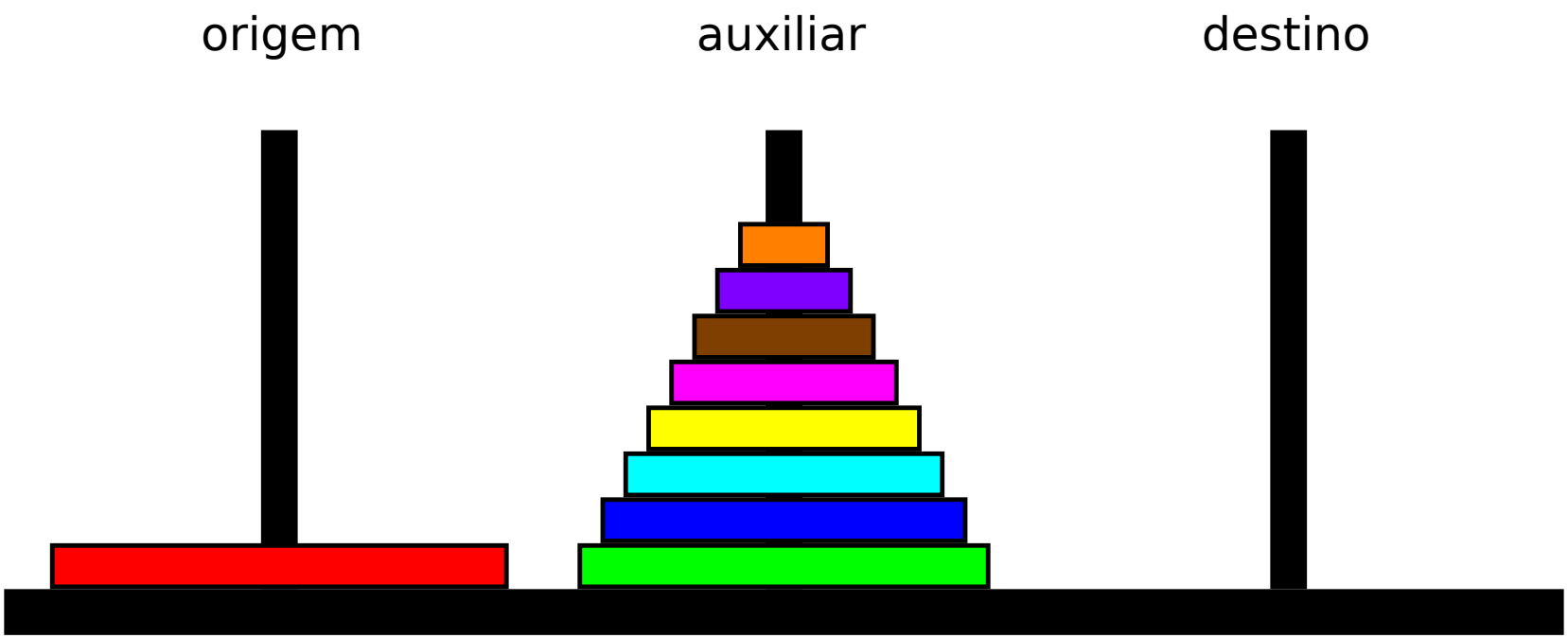
- Apenas um disco pode ser movido de cada vez.

- Um disco maior nunca pode ficar em cima de um menor.

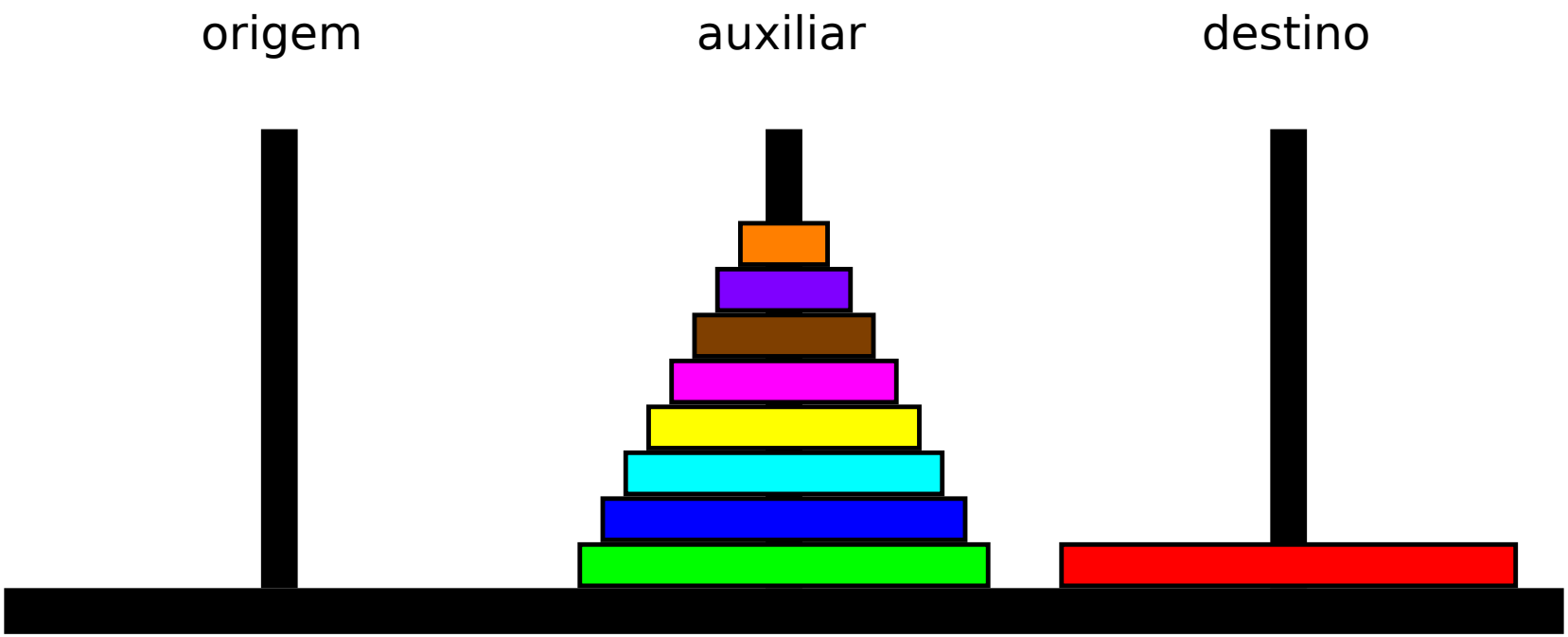
Torres de Hanói



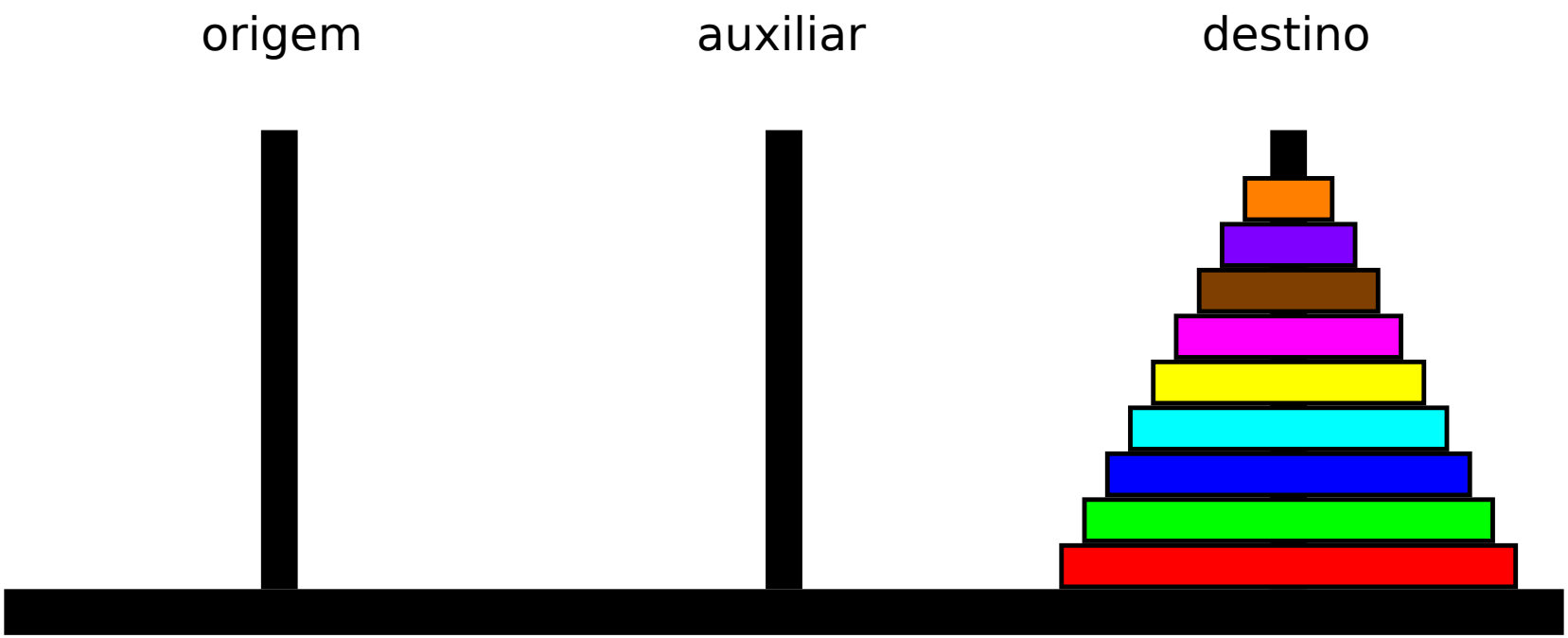
Torres de Hanói



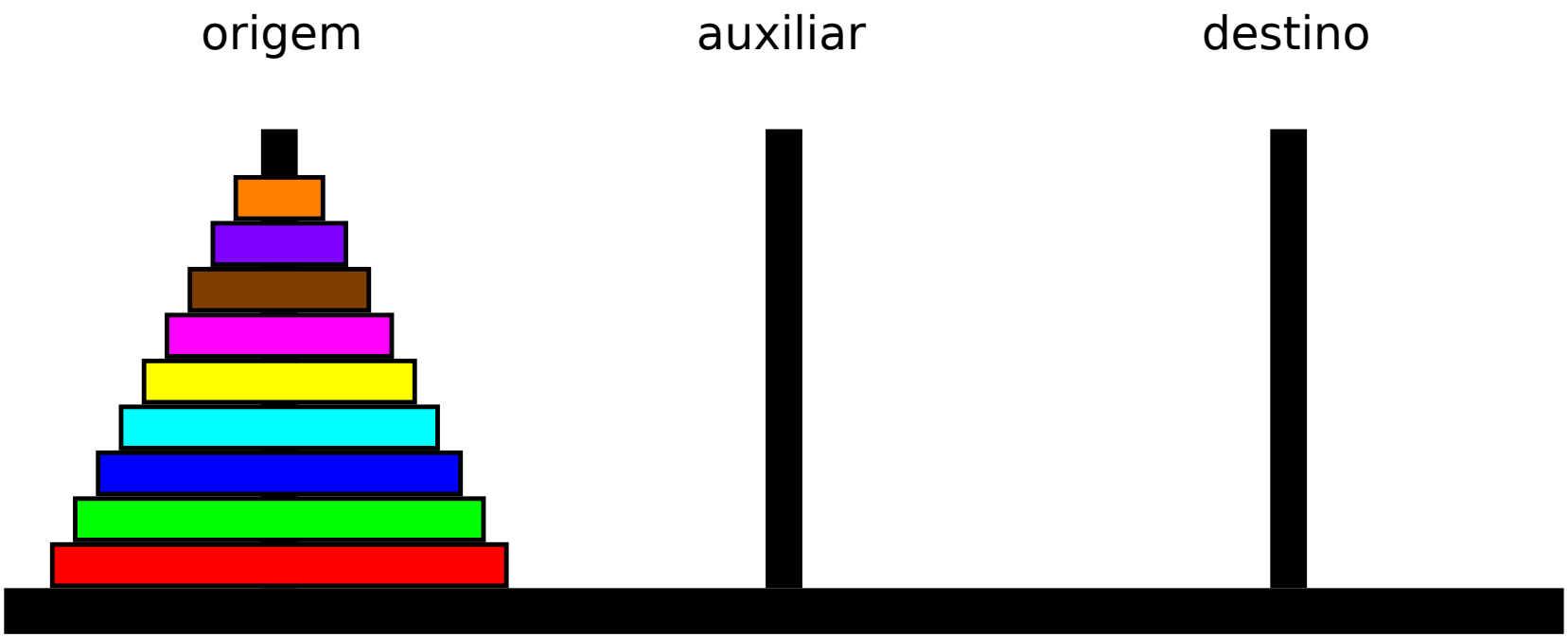
Torres de Hanói



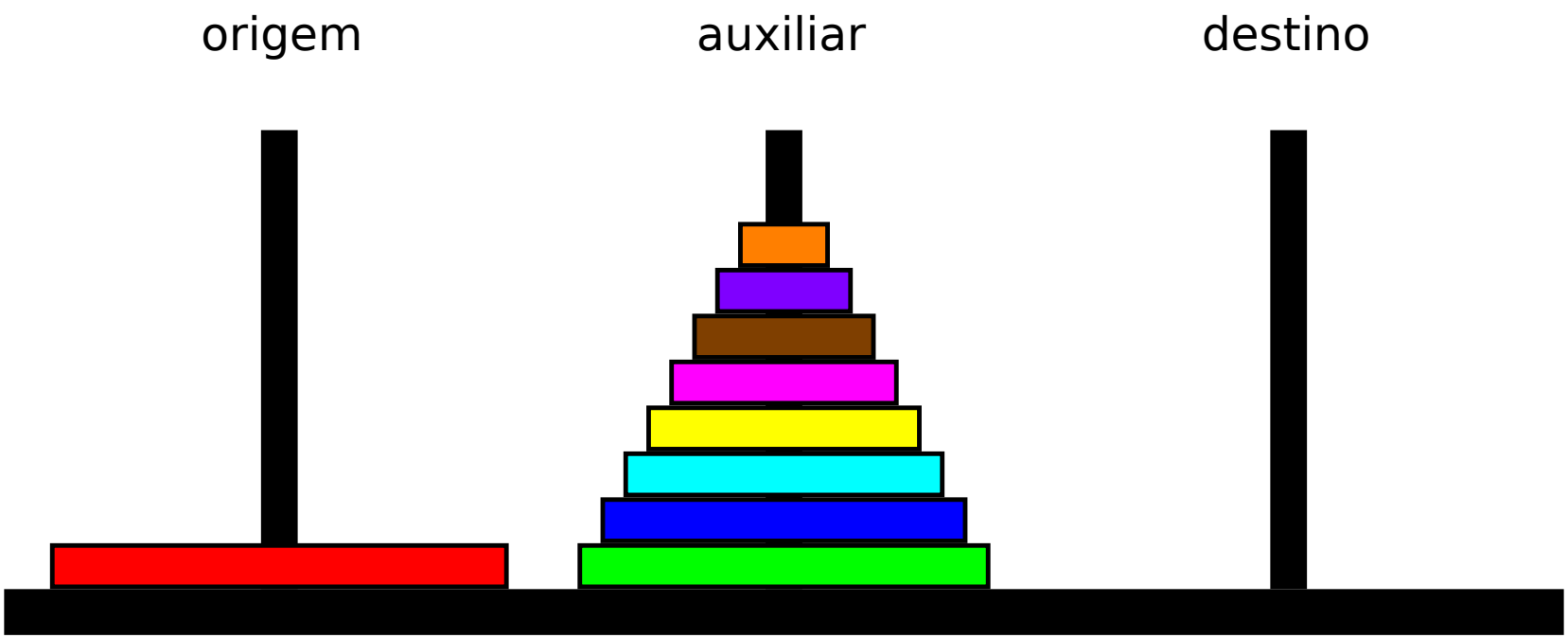
Torres de Hanói



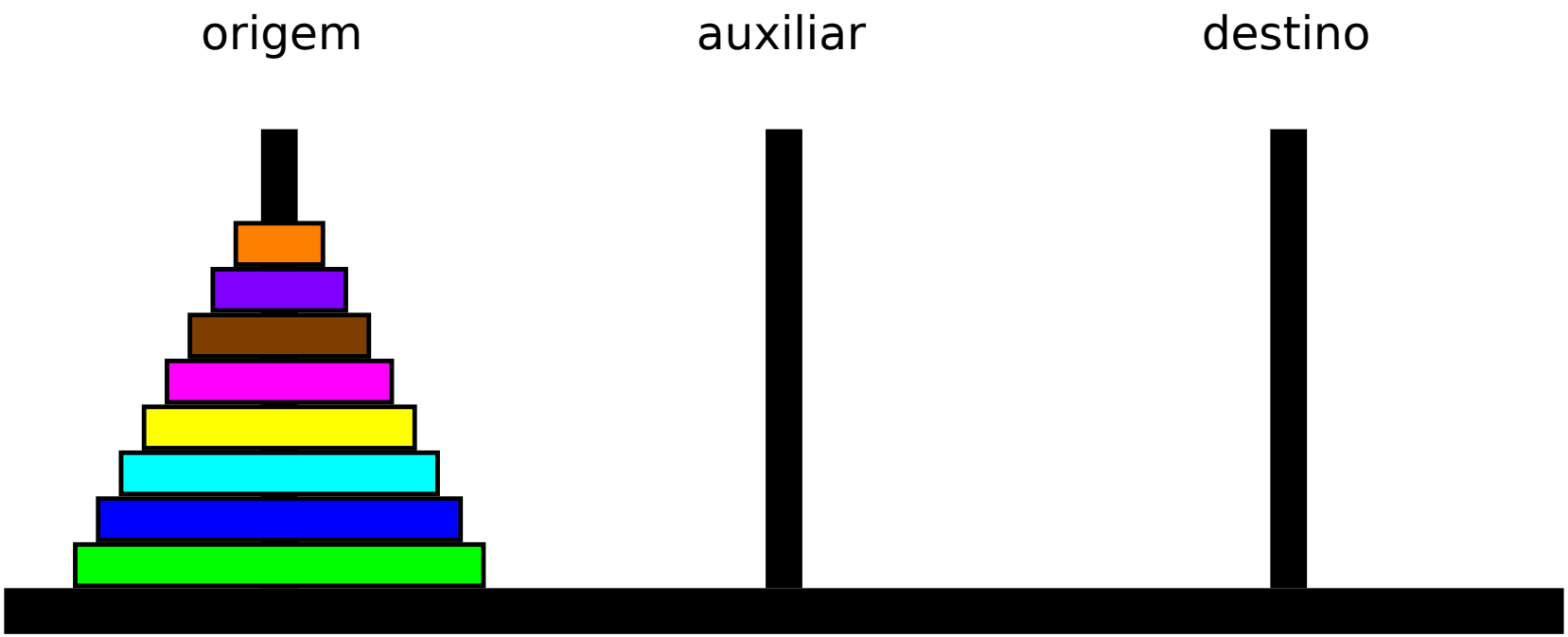
Torres de Hanói



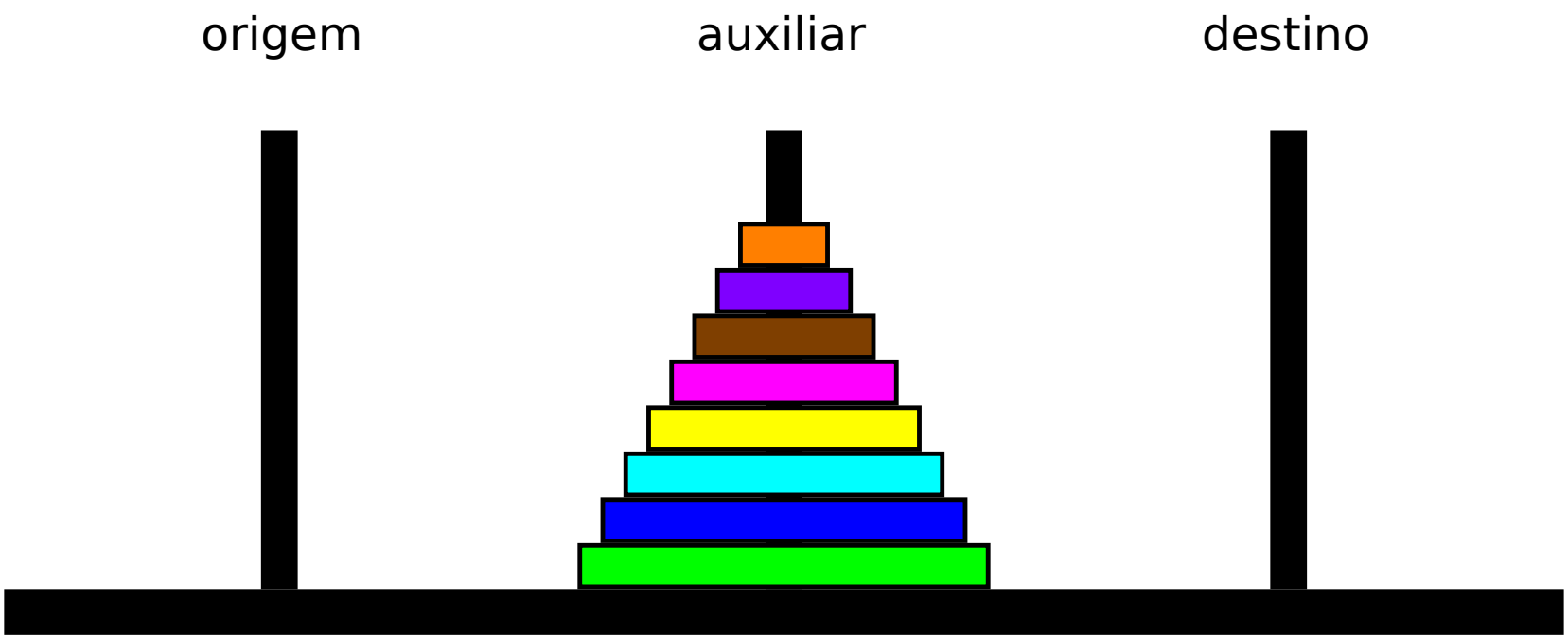
Torres de Hanói



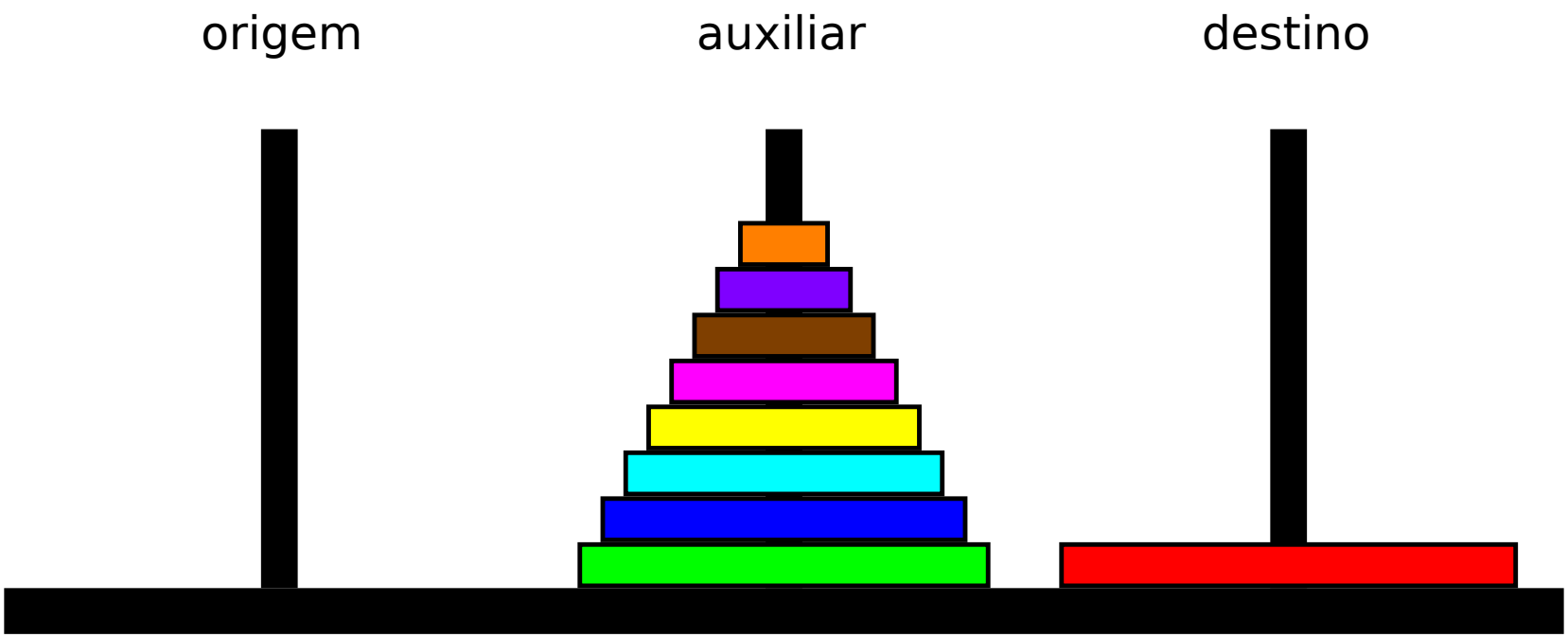
Torres de Hanói



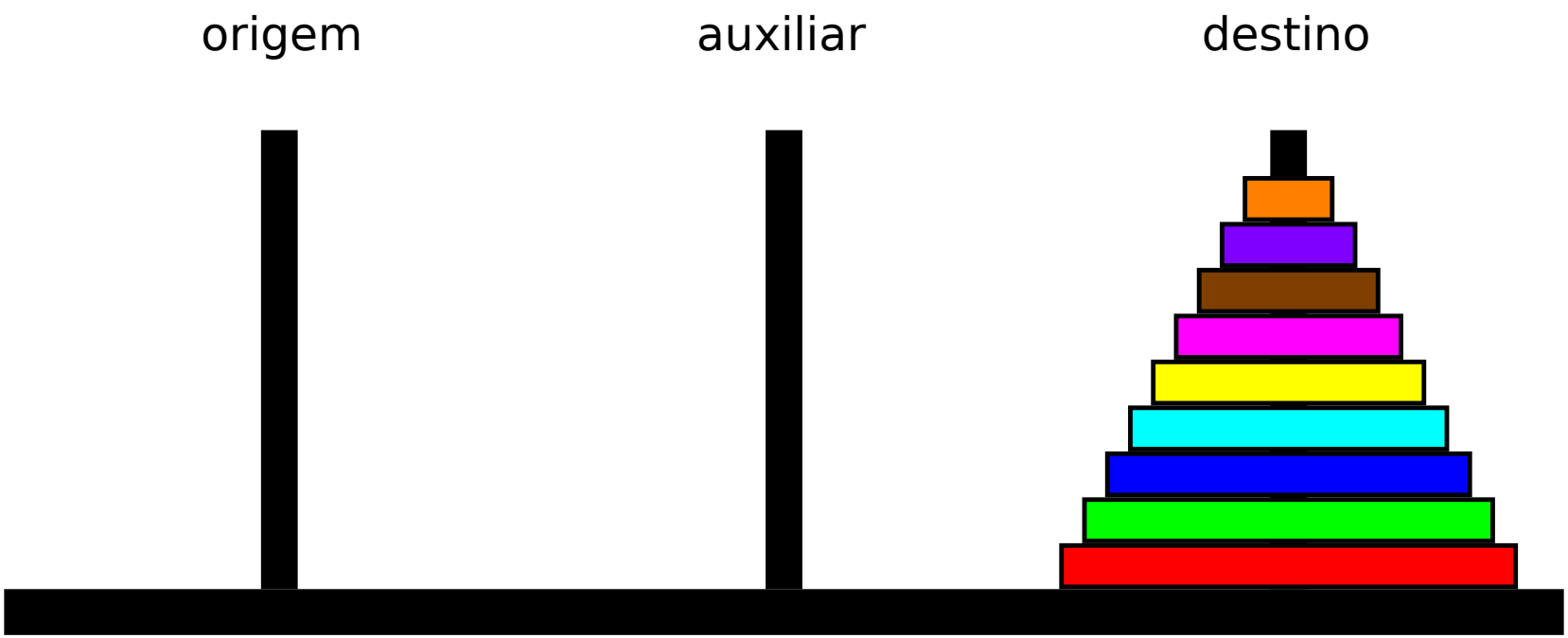
Torres de Hanói



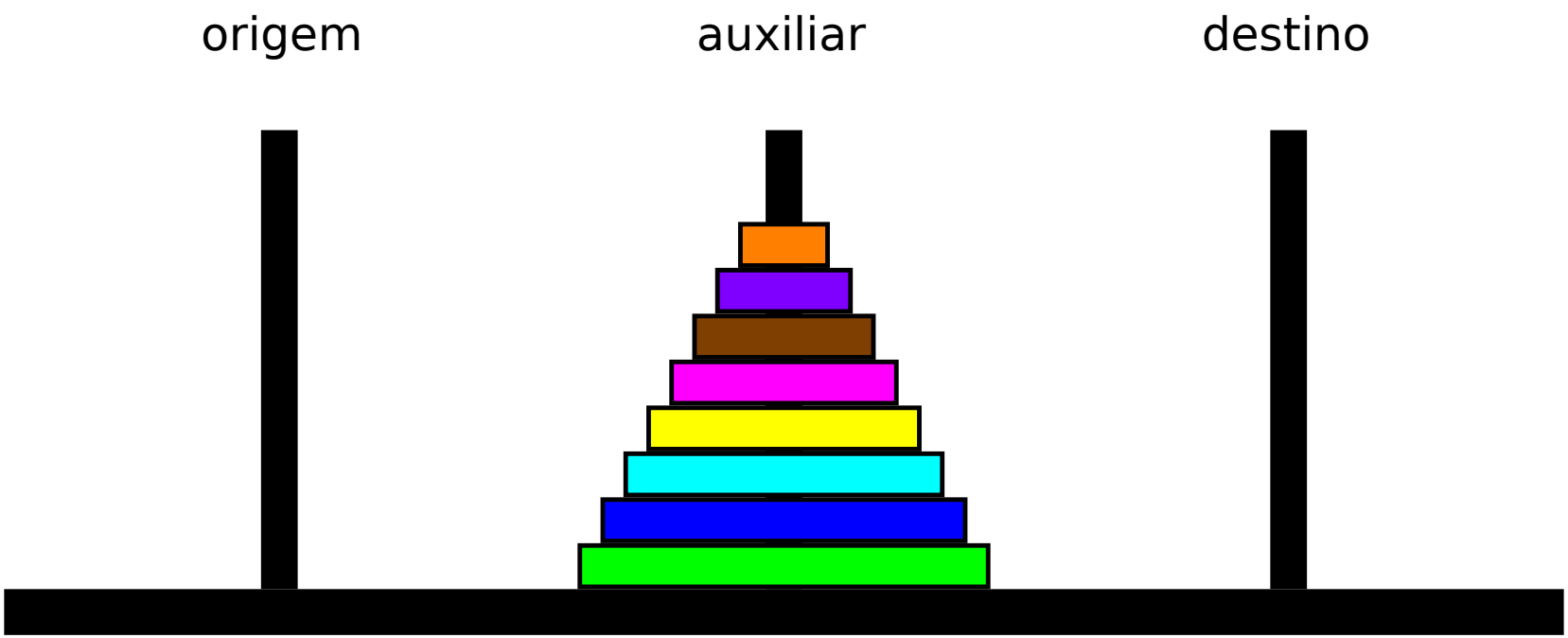
Torres de Hanói



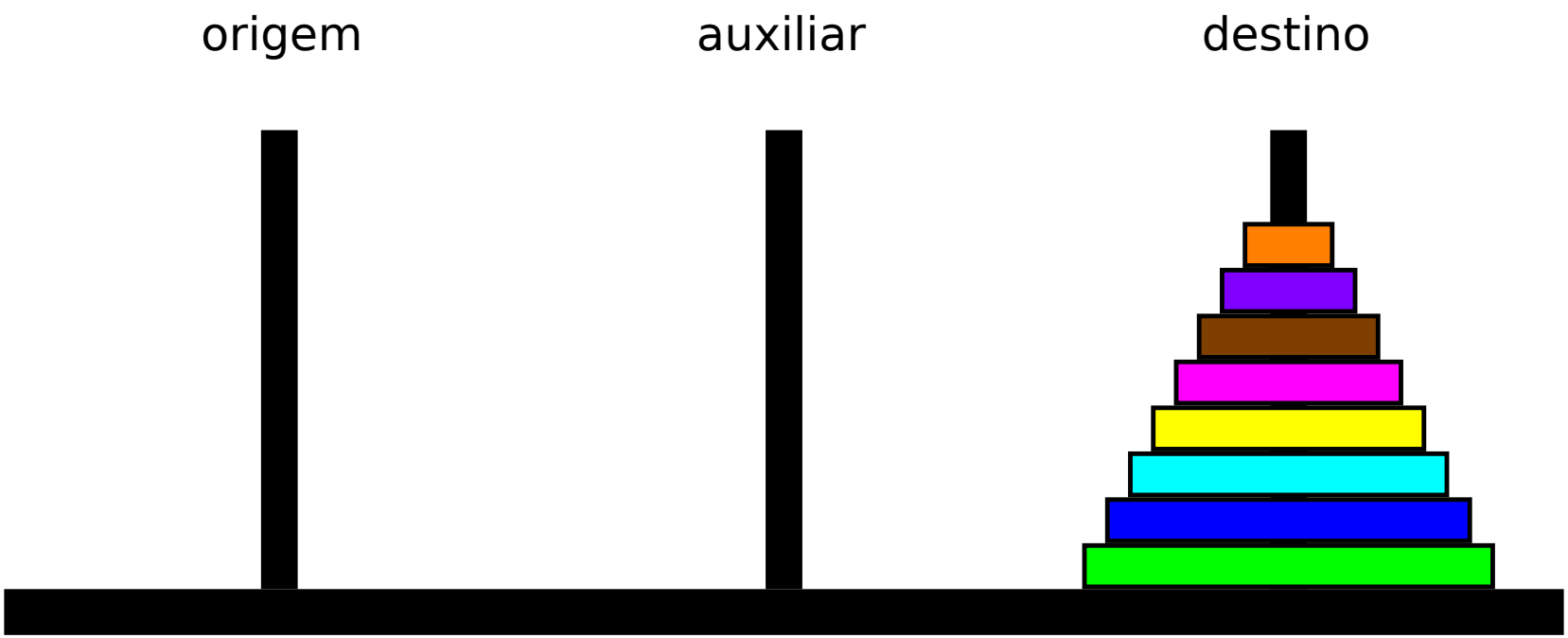
Torres de Hanói



Torres de Hanói



Torres de Hanói

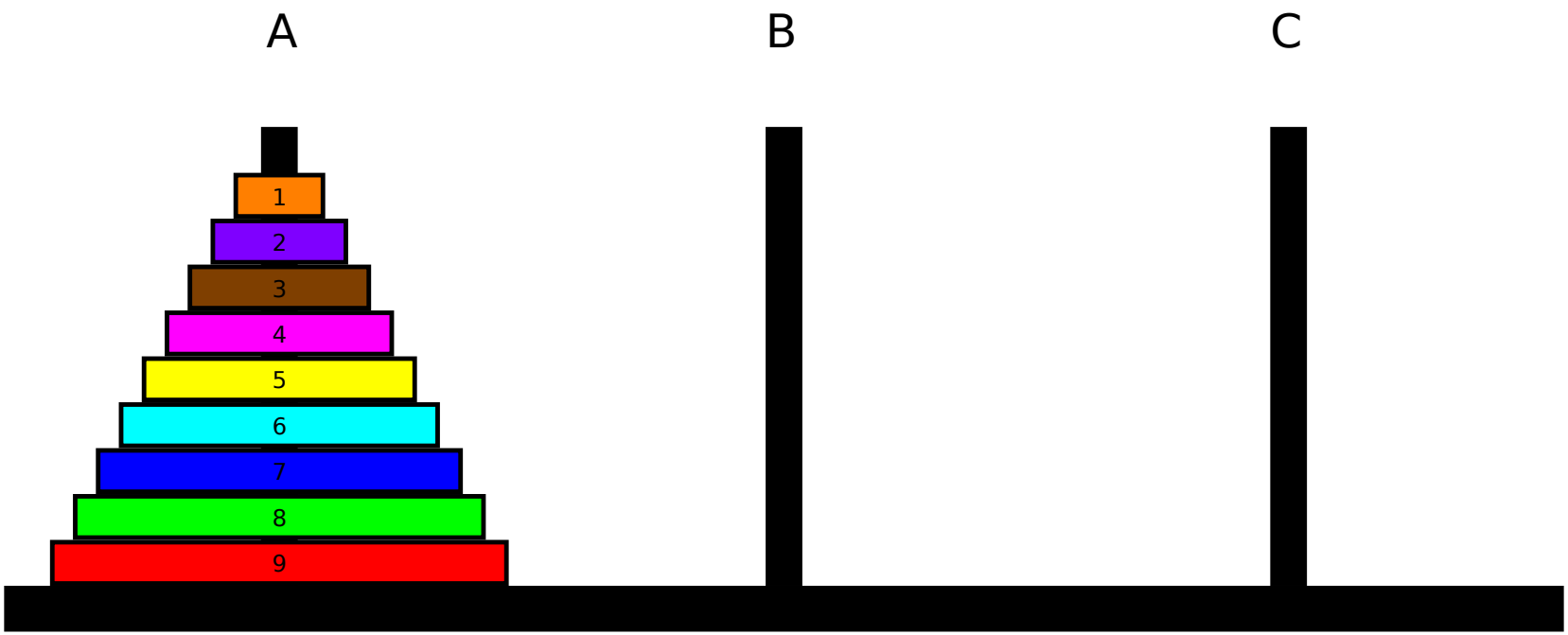


Torres de Hanói

Hanoi(n, *origem*, *auxiliar*, *destino*)

Torres de Hanói

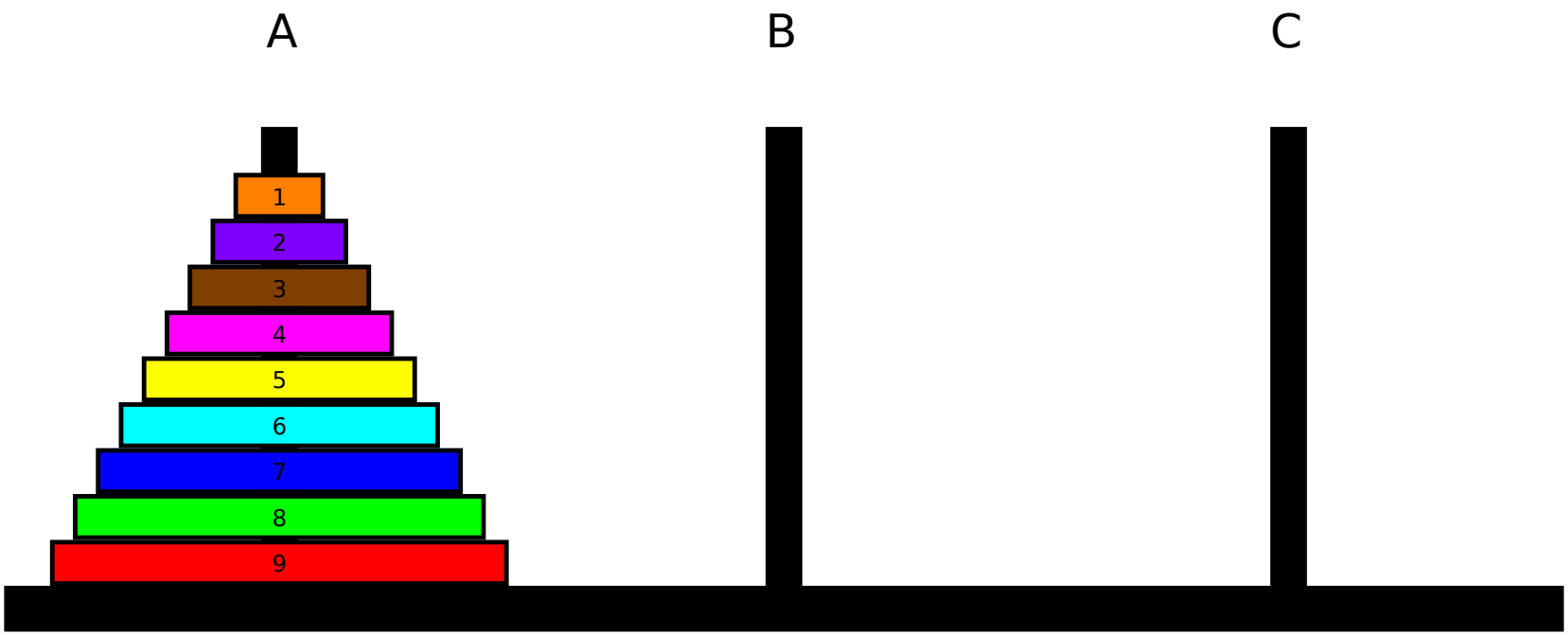
Hanoi(*n*, *origem*, *auxiliar*, *destino*)



Torres de Hanói

Hanoi(*n*, *origem*, *auxiliar*, *destino*)

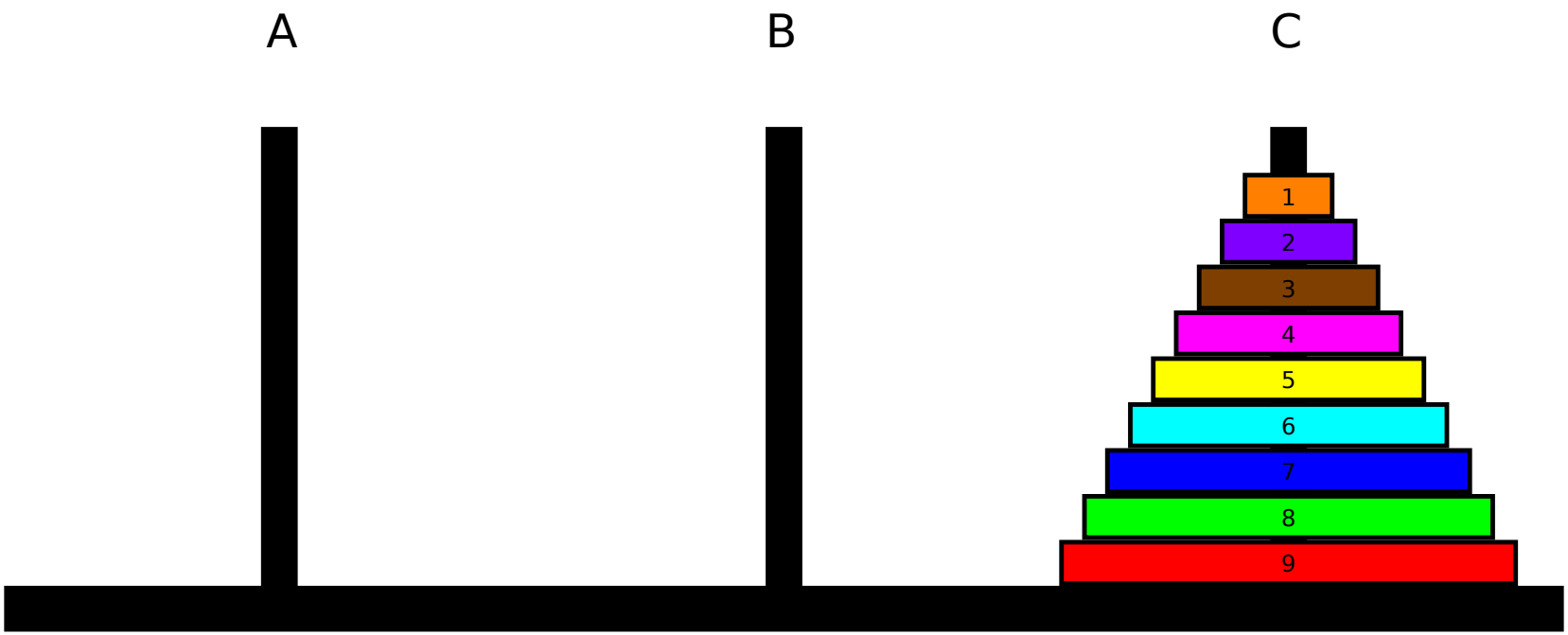
OBJETIVO: Hanoi(9, A, B, C)



Torres de Hanói

Hanoi(n, *origem*, *auxiliar*, *destino*)

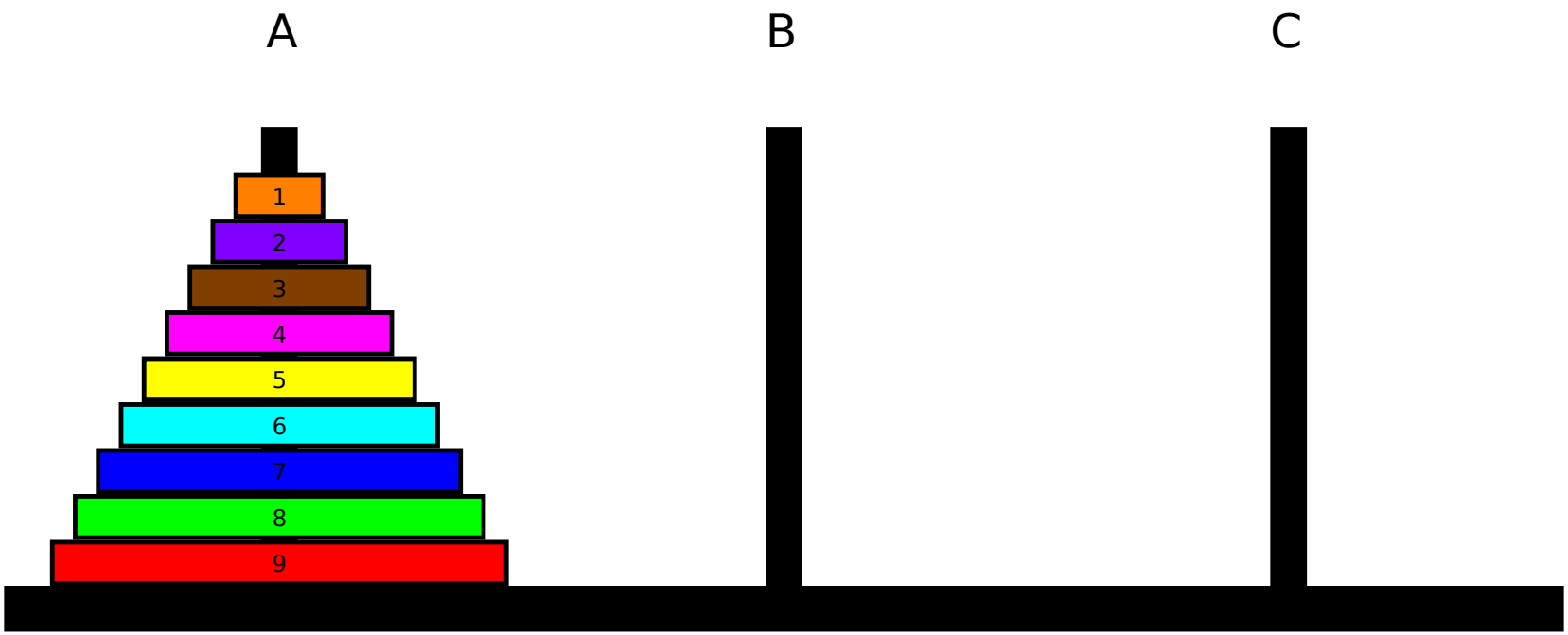
OBJETIVO: Hanoi(9, A, B, C)



Torres de Hanói

Hanoi(*n*, *origem*, *auxiliar*, *destino*)

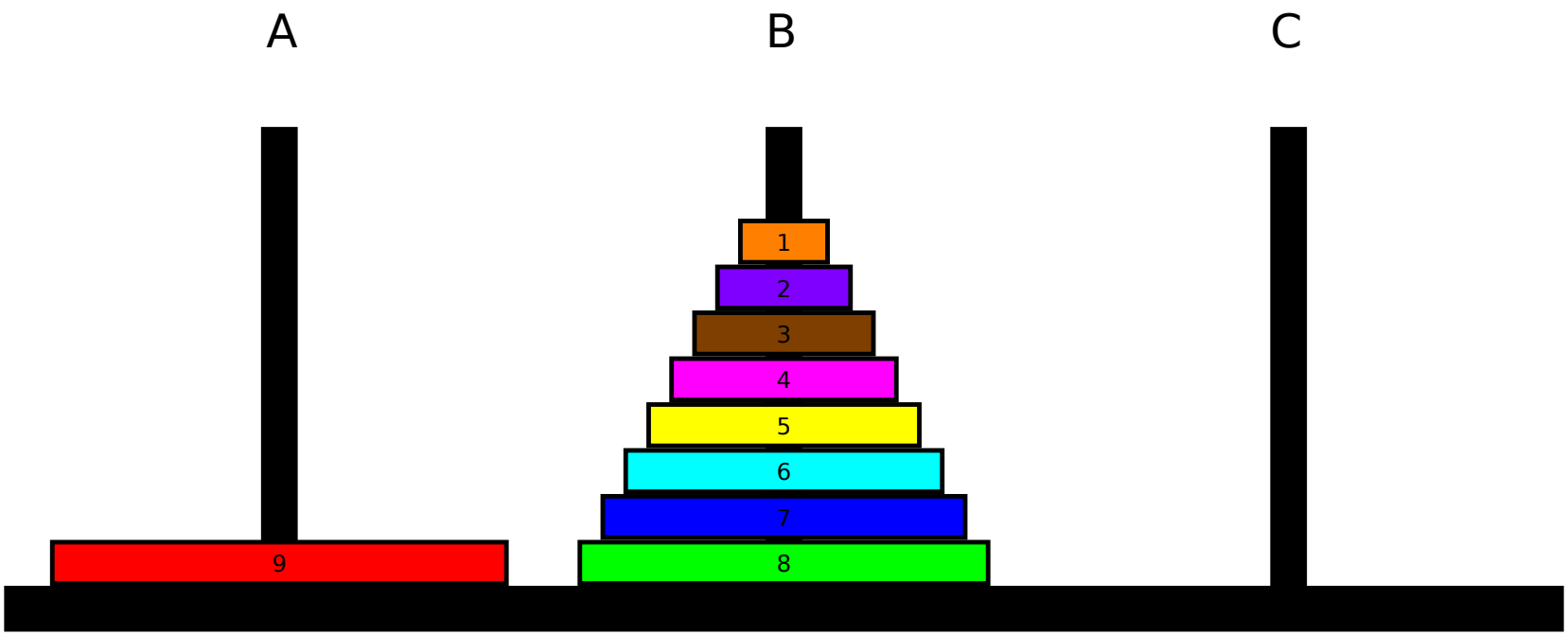
PASSO 1: chamar Hanoi(8, A, C, B)



Torres de Hanói

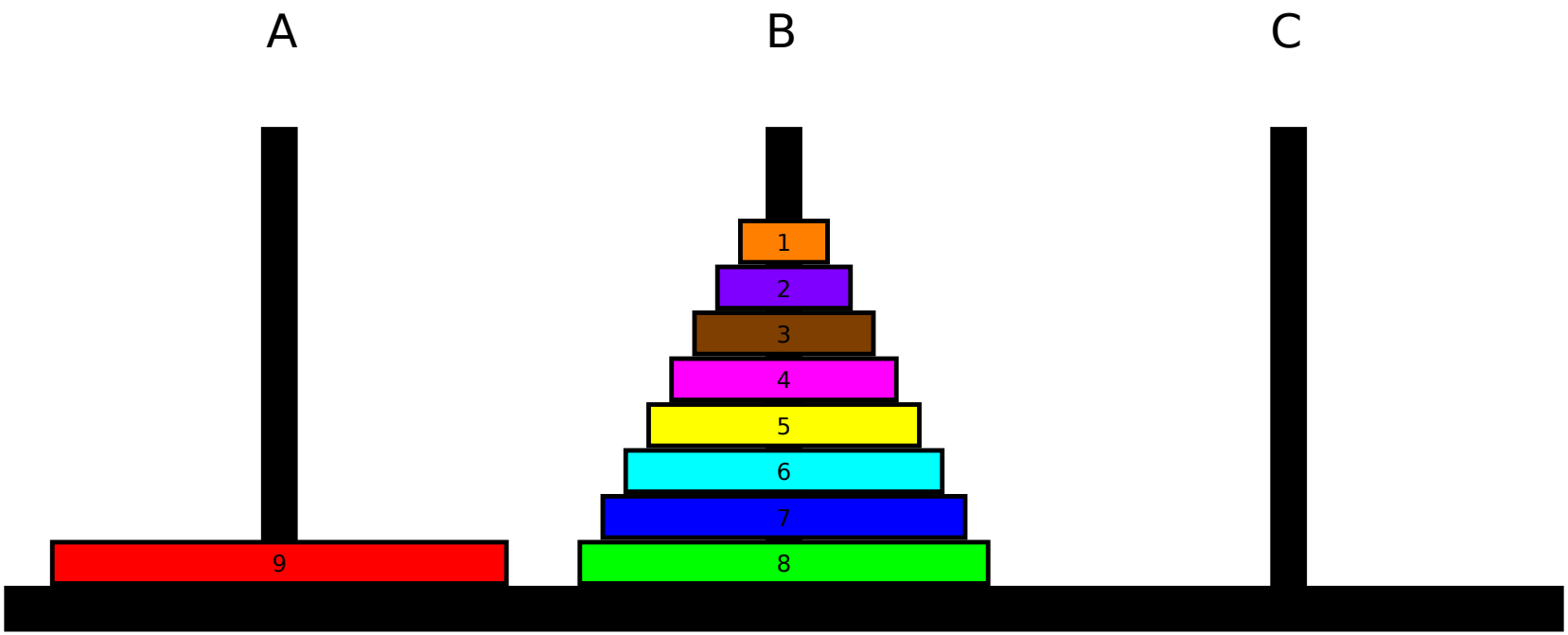
Hanoi(*n*, *origem*, *auxiliar*, *destino*)

PASSO 1: chamar Hanoi(8, A, C, B)



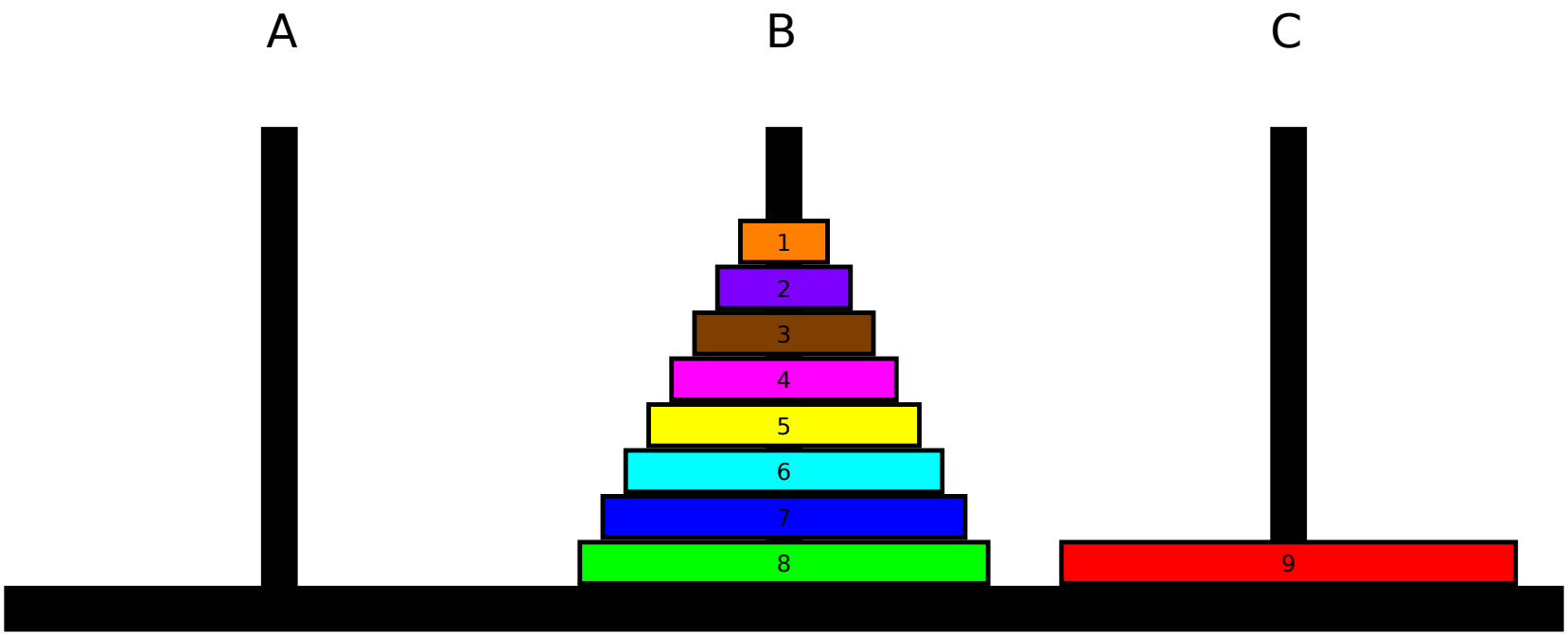
Hanoi(*n*, *origem*, *auxiliar*, *destino*)

PASSO 2: mover disco 9 de A a C



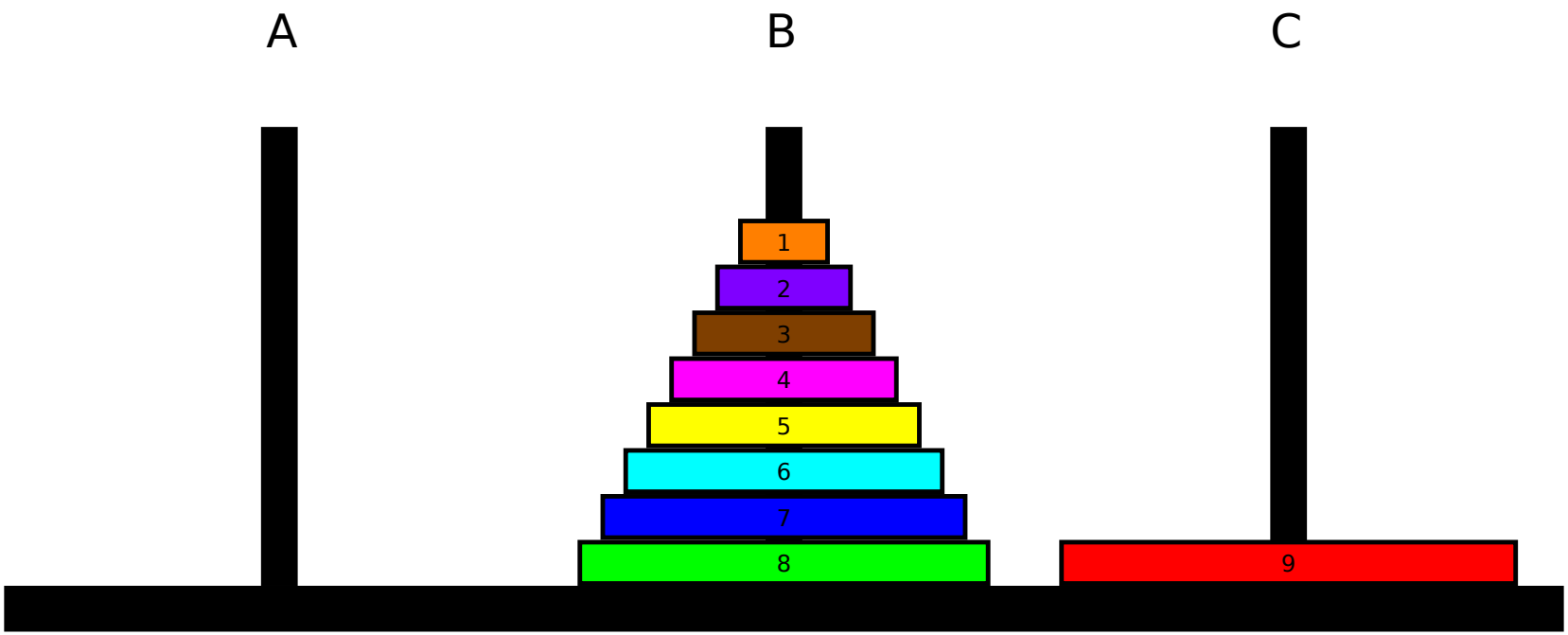
Hanoi(*n*, *origem*, *auxiliar*, *destino*)

PASSO 2: mover disco 9 de A a C



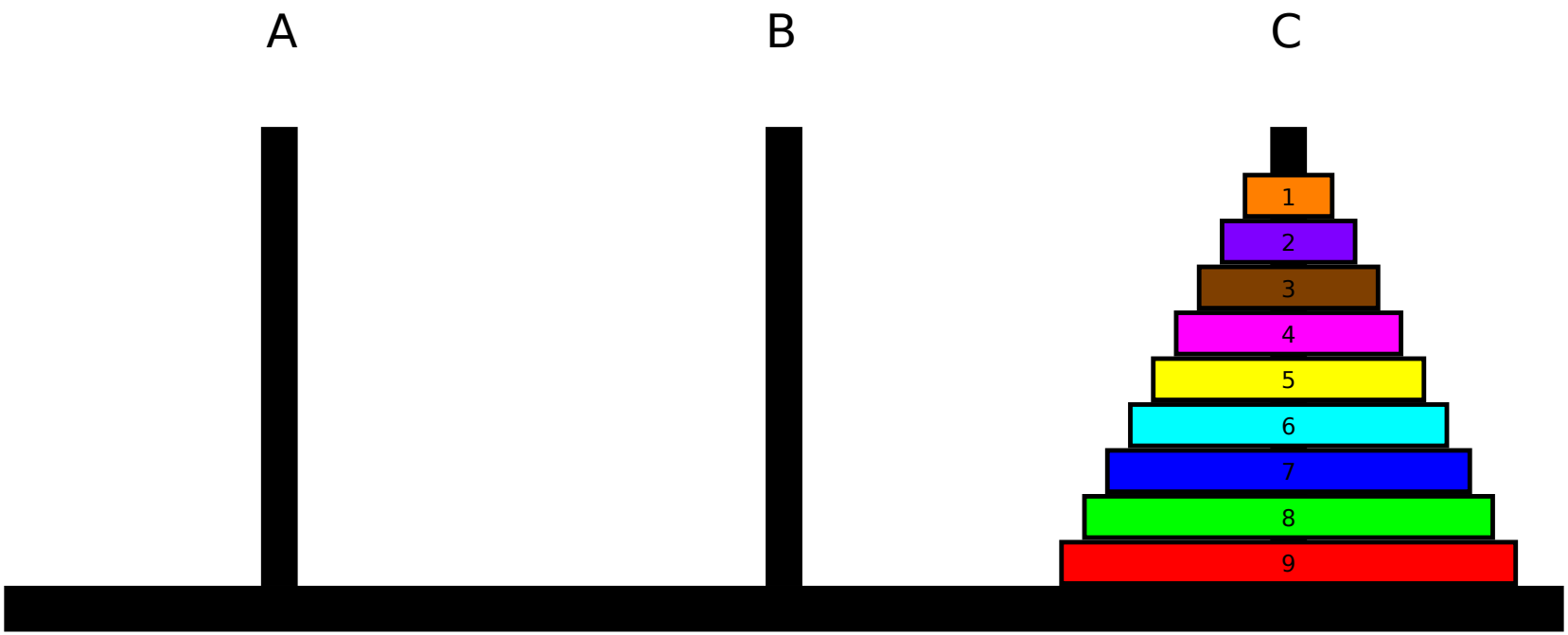
Hanoi(*n*, *origem*, *auxiliar*, *destino*)

PASSO 3: chamar Hanoi(8, B, A, C)



Hanoi(*n*, *origem*, *auxiliar*, *destino*)

PASSO 3: chamar Hanoi(8, B, A, C)



Torres de Hanói

Hanoi(9, A, B, C)

chamar Hanoi(8, A, C, B)

mover disco 9 de A a C

chamar Hanoi(8, B, A, C)

Torres de Hanói

Hanoi(*n*, *origem*, *auxiliar*, *destino*)

chamar Hanoi($n-1$, *origem*, *destino*, *auxiliar*)

mover disco n de *origem* a *destino*

chamar Hanoi($n-1$, *auxiliar*, *origem*, *destino*)

Torres de Hanói

Hanoi(n, *origem*, *auxiliar*, *destino*)

se $n > 1$
  chamar Hanoi($n - 1$, *origem*, *destino*, *auxiliar*)

mover disco n de *origem* a *destino*

se $n > 1$
  chamar Hanoi($n - 1$, *auxiliar*, *origem*, *destino*)

Exemplo

Escreva uma função recursiva que calcula x^y, onde x, y são inteiros e $y \geq 0$.

Exemplo

x^y

Exemplo

$$x^y = x \cdot x^{y-1}$$

Exemplo

$$x^y = x \cdot x^{y-1}$$
$$x^0 = 1$$

Exemplo

Potencia(x, y)

se $y = 0$
devolve 1

devolve $x \cdot$Potencia(x, $y - 1$)

Exercício

Escreva uma função recursiva que calcula $n!$, onde $n \geq 0$.

Exercício

$n!$

Exercício

$$n! = n \cdot (n-1)!$$

Exercício

$$n! = n \cdot (n-1)!$$
$$0! = 1$$

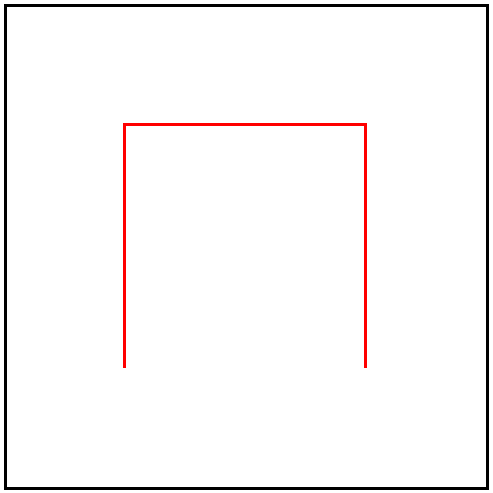
Exemplo

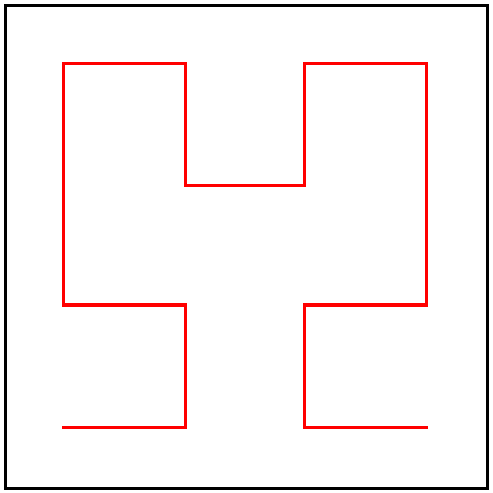
Fatorial(n)

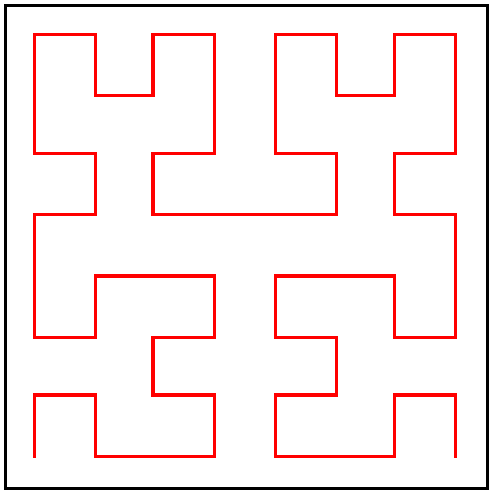
se $n = 0$
  devolve 1

devolve $n \cdot$Fatorial$(n - 1)$

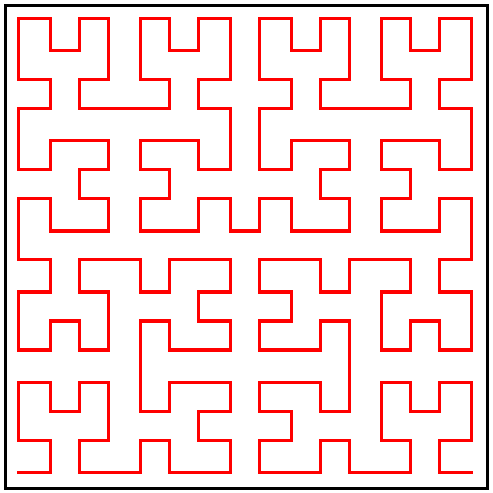
Desenhos recursivos

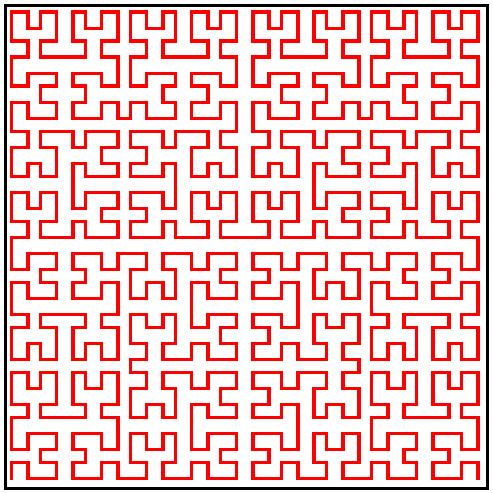


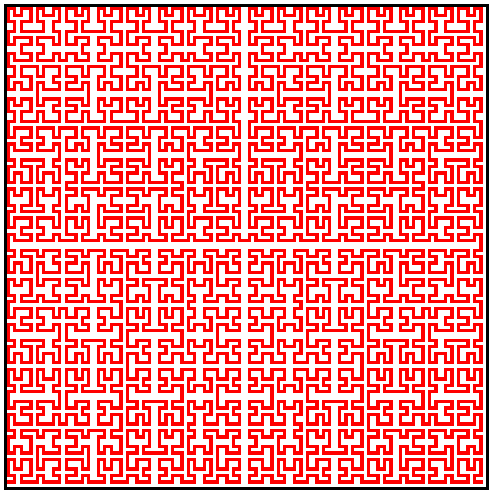




Desenhos recursivos







Torres de Hanói

Hanoi(n, *origem*, *auxiliar*, *destino*)

se $n > 1$
  chamar Hanoi($n - 1$, *origem*, *destino*, *auxiliar*)

mover disco n de *origem* a *destino*

se $n > 1$
  chamar Hanoi($n - 1$, *auxiliar*, *origem*, *destino*)

Torres de Hanói

Número de movimentos: $2^n - 1$

Torres de Hanói

Número de movimentos: $2^{64} - 1$

Torres de Hanói

Número de movimentos: 18,446,744,073,709,551,615

Supondo um movimento por segundo,
leva quase 585 bilhões de anos.