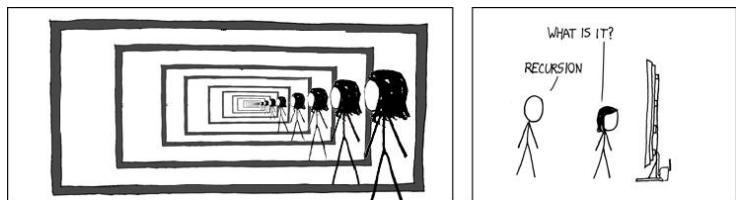


# Recursão



Fonte: <http://xkcdsw.com/1105>

PF 2.1, 2.2, 2.3 S 5.1 MD *Recursion*

<http://www.ime.usp.br/~pf/algoritmos/aulas/recu.html>

Recursão

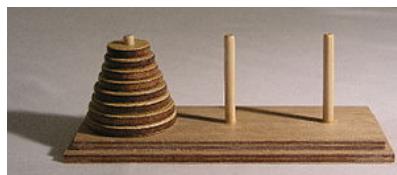
“To understand recursion, we must first understand recursion.”

-folklore

“Para fazer uma função recursivo é preciso ter fé.”

–Siang Wu Song

## Torres de Hanoi

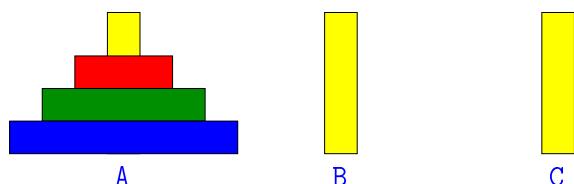


Fonte: <http://commons.wikimedia.org/>  
Licensed under Creative Commons Attribution  
Share Alike 3.0 via Wikimedia Commons

[http://en.wikipedia.org/wiki/Hanoi\\_tower](http://en.wikipedia.org/wiki/Hanoi_tower)

## Problem Solving: Tower of Hanoi

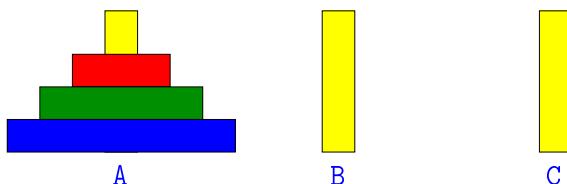
Torres de Hanoi



Desejamos transferir  $n$  discos do pino **A** para o pino **C** usando o pino **B** como auxiliar e respeitando as regras:

- ▶ podemos mover apenas um disco por vez;
  - ▶ nunca um disco de diâmetro maior poderá ser colocado sobre um disco de diâmetro menor.

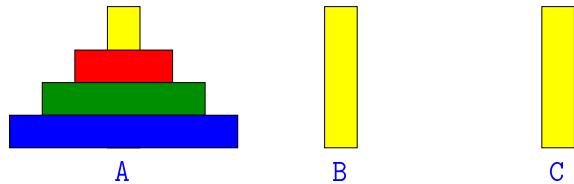
## Torres de Hanoi



Denotaremos por **Hanoi(n,A,B,C)** o problema de transferir **n** discos do pino **A** para o pino **C** usando o pino **B** como auxiliar

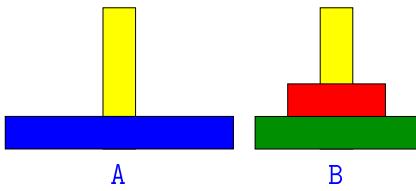
## Como resolver Hanoi(n,A,B,C)?

Idéja



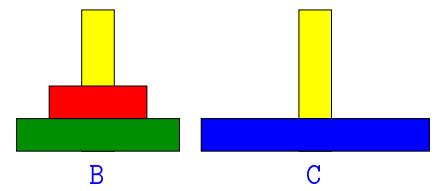
Posso não saber qual o primeiro movimento, mas é fácil saber qual é o **movimento do meio**.

## Idéia



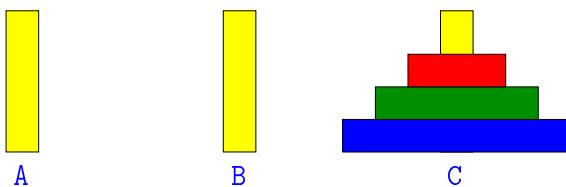
Posso não saber qual o primeiro movimento, mas é fácil saber qual é o **movimento do meio**.

## Idéia



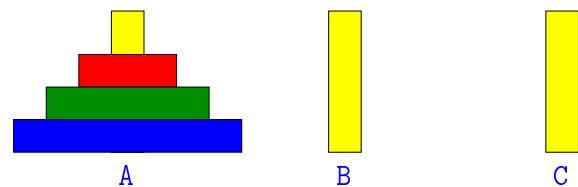
Posso não saber qual o primeiro movimento, mas é fácil saber qual é o **movimento do meio**.

## Idéia



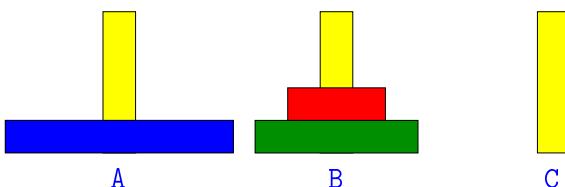
Posso não saber qual o primeiro movimento, mas é fácil saber qual é o **movimento do meio**.

## Solução



Para resolver  $\text{Hanoi}(n, A, B, C)$  basta:

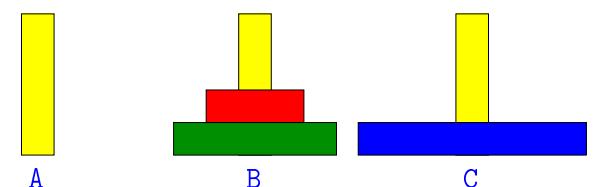
## Solução



Para resolver  $\text{Hanoi}(n, A, B, C)$  basta:

1. resolver  $\text{Hanoi}(\underline{n-1}, A, C, B)$

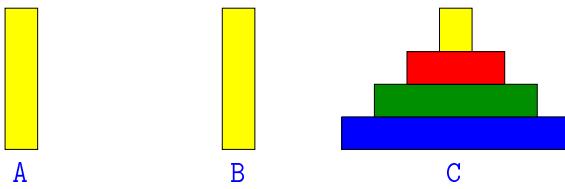
## Solução



Para resolver  $\text{Hanoi}(n, A, B, C)$  basta:

1. resolver  $\text{Hanoi}(\underline{n-1}, A, C, B)$
2. mover o disco  $n$  de  $A$  para  $C$

## Solução



Para resolver  $\text{Hanoi}(n, A, B, C)$  basta:

1. resolver  $\text{Hanoi}(n-1, A, C, B)$
2. mover o disco  $n$  de  $A$  para  $C$
3. resolver  $\text{Hanoi}(n-1, B, A, C)$

## Solução

Para resolver  $\text{Hanoi}(n, A, B, C)$  basta:

1. resolver  $\text{Hanoi}(n-1, A, C, B)$
2. mover o disco  $n$  de  $A$  para  $C$
3. resolver  $\text{Hanoi}(n-1, B, A, C)$

E dai?



## Solução

Para resolver  $\text{Hanoi}(n, A, B, C)$  basta:

1. resolver  $\text{Hanoi}(n-1, A, C, B)$
2. mover o disco  $n$  de  $A$  para  $C$
3. resolver  $\text{Hanoi}(n-1, B, A, C)$

E dai?

Reduzimos o problema com  $n$  discos para 2 problemas com  $n-1$  disco!



## Função que resolve o problema

```
def hanoi(n, origem, auxiliar, destino):  
    '''(int,str,str,str) -> None  
    Recebe um inteiro n e pinos origem, auxiliar  
    e destino e imprime os movs que resolvem  
    o problema das torres de Hanoi com n discos  
    '''  
    if n > 0:  
        hanoi(n-1, origem, destino, auxiliar)  
        print("mova disco %d de %s para %s."  
              %(n, origem, destino))  
        hanoi(n-1, auxiliar, origem, destino)
```

Primeira chamada:  $\text{hanoi}(n, 'A', 'B', 'C');$

## Solução

Para resolver  $\text{Hanoi}(n, A, B, C)$  basta:

1. resolver  $\text{Hanoi}(n-1, A, C, B)$
2. mover o disco  $n$  de  $A$  para  $C$
3. resolver  $\text{Hanoi}(n-1, B, A, C)$

E dai?

Reduzimos o problema com  $n$  discos para 2 problemas com  $n-1$  disco!

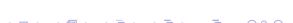
Paramos de reduzir quando soubermos resolver o problema. Por exemplo, sabemos resolver

$\text{Hanoi}(0, \dots, \dots, \dots)$



$\text{hanoi}(3, 'A', 'B', 'C')$

```
1: mova o disco 1 do pino A para o pino C.  
2: mova o disco 2 do pino A para o pino B.  
3: mova o disco 1 do pino C para o pino B.  
4: mova o disco 3 do pino A para o pino C.  
5: mova o disco 1 do pino B para o pino A.  
6: mova o disco 2 do pino B para o pino C.  
7: mova o disco 1 do pino A para o pino C.
```



## hanoi(4, A, B, C)

```
1: mova o disco 1 do pino A para o pino B.  
2: mova o disco 2 do pino A para o pino C.  
3: mova o disco 1 do pino B para o pino C.  
4: mova o disco 3 do pino A para o pino B.  
5: mova o disco 1 do pino C para o pino A.  
6: mova o disco 2 do pino C para o pino B.  
7: mova o disco 1 do pino A para o pino B.  
8: mova o disco 4 do pino A para o pino C.  
9: mova o disco 1 do pino B para o pino C.  
10: mova o disco 2 do pino B para o pino A.  
11: mova o disco 1 do pino C para o pino A.  
12: mova o disco 3 do pino B para o pino C.  
13: mova o disco 1 do pino A para o pino B.  
14: mova o disco 2 do pino A para o pino C.  
15: mova o disco 1 do pino B para o pino C.
```



## As três leis da recursão

Todos os algoritmos recursivos **devem obedecer a três leis**:

1. um algoritmo recursivo **deve** ter um **caso base**;
2. um algoritmo recursivo **deve** alterar o seu estado de maneira a **se aproximar do caso base**;
3. um algoritmo recursivo **deve chamar a si mesmo** (direta ou indiretamente).



## Curiosidades

Veja “Debugging recursive code” em :

<http://devopsreactions.tumblr.com/>

## hanoi(7, A, B, C)

```
1: mova o disco 1 do pino A para o pino B.  
2: mova o disco 2 do pino A para o pino C.  
3: mova o disco 1 do pino C para o pino A.  
4: mova o disco 3 do pino A para o pino B.  
5: mova o disco 1 do pino B para o pino A.  
6: mova o disco 2 do pino B para o pino C.  
7: mova o disco 1 do pino A para o pino C.  
8: mova o disco 4 do pino A para o pino B.  
9: mova o disco 1 do pino C para o pino A.  
10: mova o disco 2 do pino C para o pino A.  
11: mova o disco 1 do pino B para o pino C.  
12: mova o disco 3 do pino C para o pino B.  
13: mova o disco 1 do pino A para o pino C.  
14: mova o disco 2 do pino A para o pino B.  
15: mova o disco 1 do pino C para o pino A.  
16: mova o disco 3 do pino B para o pino C.  
17: mova o disco 1 do pino A para o pino B.  
18: mova o disco 2 do pino B para o pino C.  
19: mova o disco 1 do pino A para o pino B.  
20: mova o disco 3 do pino C para o pino B.  
21: mova o disco 1 do pino A para o pino C.  
22: mova o disco 2 do pino C para o pino B.  
23: mova o disco 1 do pino B para o pino C.  
24: mova o disco 4 do pino B para o pino C.  
25: mova o disco 1 do pino A para o pino C.  
26: mova o disco 2 do pino C para o pino B.  
27: mova o disco 1 do pino C para o pino A.  
28: mova o disco 3 do pino A para o pino C.  
29: mova o disco 1 do pino B para o pino A.  
30: mova o disco 2 do pino C para o pino A.  
31: mova o disco 1 do pino A para o pino C.  
32: mova o disco 6 do pino A para o pino B.  
33: mova o disco 1 do pino C para o pino A.  
34: mova o disco 2 do pino B para o pino C.  
35: mova o disco 1 do pino B para o pino A.
```

## Recursão

A resolução recursiva de um problema tem tipicamente a seguinte estrutura:

**se** a instância em questão é “**pequena**”  
**resolva-a diretamente**  
(use força bruta se necessário);  
**senão**  
reduza-a a uma instância “**menor**”  
do **mesmo problema**,  
aplique o método à instância menor e  
volte à instância original.

