MAC0115 - Introdução à Computação para Ciências Exatas e Tecnologia

Alfredo Goldman e Beatriz Viana

Setembro de 2024

EP 1

Torres de Hanoi

Neste EP você deverá desenvolver uma solução para o problema das Torres de Hanoi.

O problema da Torre de Hanoi foi inventado pelo matemático francês Edouard Lucas em 1883. Ele foi inspirado por uma lenda que fala de um templo Hindu onde o problema foi apresentado aos jovens sacerdotes. No inicio dos tempos, os sacerdotes receberam três pinos e uma pilha de 64 discos de ouro, sendo cada disco um pouco menor do que aquele abaixo dele. A tarefa era transferir todos os 64 discos colocados em um dos três pinos para outro, com duas restrições importantes. Eles só podiam mover um disco de cada vez, e eles nunca poderiam colocar um disco maior em cima de um disco menor.

O número de movimentos necessários para mover corretamente uma torre de 64 discos é $2^{64} - 1$, uma tarefa que, se movimentássemos um disco por segundo, levaria 584.942.417.355 anos para terminar.

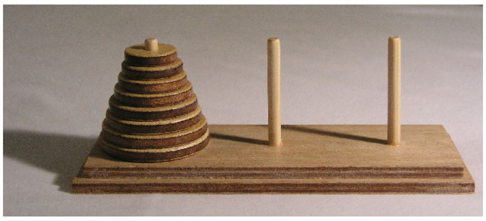

Figura 1: Torres de Hanoi com 8 discos

Esse quebra-cabeça é um exemplo clássico de problema recursivo. Para entender seu funcionamento, consideramos o problema com n discos. Enumere os discos de 1 a n, onde o disco 1 é o menor e o disco n é o maior, e nomeie os pinos da esquerda para a direita como A, B e C.

Para mover todos os discos do pino de origem, A, para o pino de destino, C, utilizando o pino B como auxiliar, sem violar as regras e utilizando a menor quantidade de movimentos, o procedimento é o seguinte:

a. Mover os discos $1, 2, \cdots, n-1$, que estão sobre n, do pino A para o pino B, usando o pino C como auxiliar; em seguida,

b. Devemos mover o disco n para o pino C, sua posição final, e

c. Mover os discos $1, 2, \cdots, n-1$ do pino B para o pino C usando A como auxiliar.

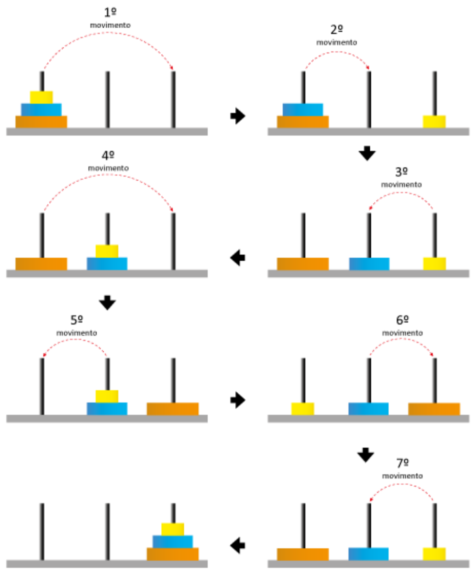

Figura 2: Esquema com 3 discos

Se abreviarmos estes movimentos como `hanoi(n, A, B, C)` podemos escrever o problema como:

a. Resolver `hanoi(n-1, A, C, B)`

b. Mover o disco n para C; e

c. Resolver `hanoi(n-1, B, A, C)`

Portanto, para resolver o problema com 4 discos, você precisará resolver `hanoi` com 3 discos duas vezes. Cada chamada para `hanoi` com 3 discos, por sua vez, requer a resolução do problema com 2 discos duas vezes.

O caso base ocorre quando há apenas um disco a ser movido. Se apenas um disco está presente, ele pode ser movido diretamente do pino de origem para o pino de destino sem precisar usar o pino auxiliar. Isso é feito com uma única movimentação:

a. Se n = 1, mova o disco do pino de origem para o pino de destino.

Esta abordagem recursiva é a base para a solução eficiente do problema das Torres de Hanoi.

Requisitos

Desenvolva duas funções que resolvem o problema, `hanoi(n, dep)`, onde `n` é o número de discos do problema e `dep` é uma variável binária que indica se o modo depuração deve estar ativo. Se `dep` for igual a 1 as movimentações dos discos devem ser indicadas na tela. Além disso, a função `hanoi` deve imprimir na tela quantos movimentos foram necessários para resolver `hanoi` com `n` discos.

Além desta função, você deve desenvolver a função `hanoi_rec(n, dep, origem, auxiliar, destino)` que é a função auxiliar recursiva que de fato resolve hanoi, ou seja, ela é responsável por realizar a contagem de movimentos e as movimentações dos discos (e, portanto, a impressão dessas movimentações na tela). Esta função deve ser chamada pela função principal `hanoi(n,dep)`.

Exemplo de Uso

Podemos utilizar o código para testar o caso clássico do problema das Torres de Hanoi, ou seja, quando `n` é igual a 3 discos.

Caso você chame a função `hanoi` com o modo de depuração igual a 0, o resultado será:

```
Foram necessários 7 movimentos
```

Já se o modo de depuração for igual a 1, então o resultado será:

```
Mova o disco 1 do pino A para o pino C
Mova o disco 2 do pino A para o pino B
Mova o disco 1 do pino C para o pino B
Mova o disco 3 do pino A para o pino C
Mova o disco 1 do pino B para o pino A
Mova o disco 2 do pino B para o pino C
Mova o disco 1 do pino A para o pino C
Foram necessários 7 movimentos
```

Mais alguens exemplos de execução são:

```
$ hanoi(2, 0)
Foram precisos 3 movimentos

$ hanoi(2, 0)
Mova o disco 1 do pino A para o pino B
Mova o disco 2 do pino A para o pino C
Mova o disco 1 do pino B para o pino C
Foram precisos 3 movimentos

$ hanoi(4, 0)
Foram precisos 15 movimentos

$ hanoi(5, 0)
Foram precisos 31 movimentos

$ hanoi(6, 0)
Foram precisos 63 movimentos
```

Faça as funções exatamente como foram pedidas, sem alterar seu cabeçalho nem funcionamento. As frases que devem ser impressas na tela também não devem ser alteradas.

Entrega

Um código-fonte foi disponibilizado com o cabeçalho de cada função, você deve utilizar este código como base para desenvolver sua solução. Por fim, o código-fonte da solução deve ser submetido, com a extensão de Julia, `.jl`, ou `.ipynb` caso tenha desenvolvido o código no Google Colaboratory.

Não esqueça de fazer um código bem escrito e indentado :).

Extra

No site da OBMEP você podem jogar `hanoi` para quantos discos você escolher, isso pode ajudar a entender o funcionamento do quebra-cabeça.