Departamento de Ciência da Computação – IME-USP – Segundo Semestre de 2015

**MAC 115 – Introdução à Computação – IF - diurno - Turma 23**

# EP3 - O Problema da Torre de Hanoi

Exercício-Programa 3 (EP3) – Data de entrega: **28 de outubro de 2015 (5a.feira)**

O seguinte trecho foi copiado do site

`http://www.lawrencehallofscience.org/java/tower/towerhistory.html`

```
Tower of Hanoi

  Fascinating Facts

       The Tower of Hanoi (sometimes referred to as the
       Tower of Brahma or the End of the World Puzzle)
       was invented by the French mathematician,              |        |       |
       François Édouard Lucas, in 1883. He was inspired       -        |       |
       by a legend that tells of a Hindu temple where        ---       |       |
       the pyramid puzzle might have been used for the      -----      |       |
       mental discipline of young priests. Legend says     *******  ******* *******
       that at the beginning of time the priests in          p1       p2      p3
       the temple were given a stack of 64 gold disks,
       each one a little smaller than the one beneath it. Their assignment was to transfer
       the 64 disks from one of the three poles to another, with one important proviso:
       a large disk could never be placed on top of a smaller one. The priests worked
       very efficiently, day and night. When they finished their work, the myth said,
       the temple would crumble into dust and the world would vanish.


  Where's the Math in this Game?

       The number of separate transfers of single disks the priests must make to
       transfer the tower is 2 to the 64th minus 1, or 18,446,744,073,709,551,615 moves!
       If the priests worked day and night, making one move every second it would take
       slightly more than 580 billion years to accomplish the job!
       You have a great deal fewer disks than 64 here (OBS: aqui sao
       dados 3 discos). Can you calculate the number of moves it will
       take you to move the disks from one of the three poles to another?
```

Veja também:

`http://www2.mtsd.k12.wi.us/Homestead/users/ordinans/Tower%0of%20Hanoi.html`

## 2. Descrição do Problema e um Algoritmo

Vamos descrever o problema mais formalmente, de modo a ter uma notação padronizada para este exercício.

Considere 3 pinos $p1$, $p2$ e $p3$ e $n$ discos $1, 2, \ldots, n$, cada disco $i$ com diâmetro $d_i$, sendo $d_1 < d_2 < \ldots < d_n$.

Inicialmente esses $n$ discos encontram-se no pino $p1$, arranjados em ordem decrescente (de diâmetro) da base até ao topo; e os pinos $p2$ e $p3$ estão sem nenhum disco (como mostra a figura da página anterior).

O problema consiste em mover todos os $n$ discos para o pino $p3$, através de uma seqüência de movimentos, obedecendo às seguintes regras:

**(R1)** Pode-se mover um único disco por vez (de um pino para outro);

**(R2)** Em nenhum momento um disco (de diâmetro) maior pode ser colocado sobre um outro (de diâmetro) menor.

Por exemplo, se $n = 3$, uma seqüência de movimentos possíveis seria: $1 \rightarrow p3$ (disco 1 vai para $p3$), $2 \rightarrow p2$ (disco 2 vai para $p2$), $1 \rightarrow p2$, $3 \rightarrow p3$, $1 \rightarrow p1$, $2 \rightarrow p3$, $1 \rightarrow p3$.

Este problema é geralmente usado para exemplificar o uso de *recursão* em cursos de programação. Neste exercício ainda não faremos uso de recursão (logo você vai aprender o que é isso). Na verdade, vamos dar um algoritmo para encontrar uma tal seqüência, e você deve simplesmente implementar esse algoritmo.

**O Algoritmo:**

- **Considere os pinos como se estivessem arranjados em um círculo, com movimentos no sentido** *horário* **sendo de** $p1 \rightarrow p2 \rightarrow p3 \rightarrow p1$**. (Assim, no sentido** *anti-horário* **os movimentos são de** $p1 \rightarrow p3 \rightarrow p2 \rightarrow p1$**.)**
- **Se** $n$ **é** *ímpar***, comece movendo o disco 1 para o próximo pino no sentido** *anti-horário***. Repetidamente, alterne os movimentos de mover o disco 1 dessa maneira, e o único outro movimento possível. Faça isso até que os discos estejam todos no pino** $p3$**.**
- **Se** $n$ **é** *par***, comece movendo o disco 1 para o próximo pino no sentido** *horário***. Repetidamente, alterne os movimentos de mover o disco 1 dessa maneira, e o único outro movimento possível. Faça isso até que os discos estejam todos no pino** $p3$**.**

OBS1: Note que, a qualquer momento, o disco 1 está sempre no topo de um dos pinos. Como ele é o menor disco, sempre é possível movê-lo para um outro pino, obedecendo assim a regra R2. O IMPORTANTE é que, o algoritmo dado impõe que *os movimentos do disco 1 devem sempre ser no sentido anti-horário (resp. horário) se n é ímpar (resp. par).* Conforme o algoritmo acima, cada movimento do disco 1 deve ser seguido de um movimento (válido) que não envolve o disco 1. Assim, se o disco 1 foi movido para um pino $k$, o próximo passo consiste em mover um disco entre os pinos $i$ e $j$, onde $i$ e $j$ são ambos distintos de $k$. Para descobrir se o movimento vai ser mover um disco do pino $i$ para o pino $j$, ou do pino $j$ para o pino $i$, basta testar qual deles é válido (ou seja, obedece a regra R2). Como apenas um deles é válido, tal movimento válido deve ser feito, voltando-se a repetir o processo (move-se o disco 1, move-se outro disco, move-se o disco 1, move-se outro disco, e assim sucessivamente.)

OBS2: O sentido anti-horário (resp. horário) para $n$ ímpar (resp. ímpar), é para garantir que os discos (que começam todos no pino 1) terminem no pino 3. (Se no caso de $n$ ímpar for usado o sentido horário, os discos vão terminar todos no pino 2.)

Para ver se você entendeu bem o algoritmo, veja um exemplo para $n = 3$ em:

`http://www2.mtsd.k12.wi.us/Homestead/users/ordinans/Tower%20of%20Hanoi.html`

Para aprender mais a respeito do Problema da Torre de Hanoi, veja a página:

`http://www.lchr.org/a/36/oh/stands/hanoi1.htm`

Nessa página você vai entender porque o número total de movimentos para resolver o problema para um dado $n$ é precisamente $2^n - 1$.

## 3. Seu programa

Você deve fazer um programa em C para resolver o seguinte problema:

a) Dado um inteiro positivo $n$ (a ser lido), $n < 8$, resolver o Problema da Torre de Hanoi para $n$ discos. Ou seja, são dados $n$ discos, digamos $n$, $n-1$, $n-2$,..., 1 no pino 1, onde disco $i$ tem diâmetro $i$, deseja-se mover esses $n$ discos para o pino 3, usando o algoritmo que foi explicado acima.

b) O seu programa deve imprimir a configuração de cada um dos pinos: pino 1, pino 2, pino 3 (nessa ordem), depois de cada movimento. (Veja o exemplo abaixo.)

c) O seu programa deve contar o total de movimentos realizados, e indicar qual é esse valor no final.

Dica: Inicialmente, faça um programa apenas para o caso $n$ par (ou ímpar). Se este estiver correto, faça para o outro caso (que é análogo).

Exemplo: se $n = 3$, o seu programa deve exibir uma saída da forma abaixo (ou algo similar)

```
n = 3 (ordem para mover o disco 1: p1--> p3 --> p2 --> p1)

p1 = 3  2  1
p2 =
p3 =

p1 = 3  2
p2 =
p3 = 1

p1 = 3
p2 = 2
p3 = 1

p1 = 3
p2 = 2  1
p3 =

p1 =
p2 = 2  1
p3 = 3

p1 = 1
p2 = 2
p3 = 3

p1 = 1
p2 =
p3 = 3  2

p1 =
p2 =
p3 = 3  2  1

Total de movimentos realizados = 7
```

OBS: Na aula veremos dicas sobre o uso de *arrays* para representar cada um dos 3 pinos e dicas de inicialização desses *arrays*. O conceito de "pilhas" (*stacks*) e um apontador para o "topo" da pilha serão também discutidos.