

# O problema das Torres de Hanoi: a lenda, algoritmos e generalizações.

António Pereira e Rosália Rodrigues  
Departamento de Matemática da Universidade de Aveiro

## 1. Origem, lendas e mitos

Em 1883, o matemático francês Édouard Lucas inventou o famoso puzzle das *Torres de Hanoi* [6] [7], também conhecido pelas *Torres de Brahma* e contado em forma de lenda [5]:

*“No grande templo de Brahma em Benares, numa bandeja de metal sob a cúpula que marca o centro do mundo, três agulhas de diamante servem de pilar a sessenta e quatro discos de ouro puro. Incansavelmente, os sacerdotes transferem os discos, um de cada vez, de agulha para agulha, obedecendo sempre à lei imutável de Brahma: Nenhum disco se poderá sobrepor a um menor.*

*No início do mundo todos os sessenta e quatro discos de ouro, foram dispostos na primeira das três agulhas, constituindo a Torre de Brahma. No momento em que o menor dos discos for colocado de tal modo que se forme uma vez mais a Torre de Brahma numa agulha diferente da inicial, tanto a torre como o templo serão transformados em pó e o ribombar de um trovão assinalará o fim do mundo.”*

## 2. O Algoritmo Recorrente

A versão original das *Torres de Hanoi* consiste em três postes e oito discos de diâmetro  $1, 2, \dots, 8$ , inicialmente dispostos no primeiro poste por ordem decrescente do diâ-

metro formando uma estrutura cónica semelhante à da figura 1. O objectivo do puzzle consiste em formar a torre no terceiro poste, movendo um disco de cada vez, não sendo permitido colocar um disco maior sobre um menor.



Figura 1: A configuração inicial das Torres de Hanoi com 8 discos.

### Problema 2.1 Torres de Hanoi

São dados  $n$  discos de diâmetro  $1, 2, \dots, n$  dispostos por ordem decrescente de diâmetro num de 3 postes. Pretende-se transferir todos os discos para um dos outros postes, utilizando o menor número de movimentos, de tal modo que as seguintes restrições sejam satisfeitas:

1. apenas um disco pode ser movido de cada vez,
2. apenas se podem mover os discos do topo (isto é, apenas discos que não têm um outro disco colocado em cima),
3. nenhum disco pode ser colocado sobre outro menor.

A solução para o problema 2.1 pode ser descrita da seguinte forma:

Designem-se por *poste inicial* aquele onde inicialmente se encontram os  $n$  discos, por *poste final* o poste para onde serão transferidos os discos e por *poste auxiliar* o poste restante.

Para transferir  $n$  discos de um poste para outro é necessário, em alguma iteração, mover o disco  $n$  pelo menos

uma vez. Este só pode ser transferido se for o único no poste e existir um poste vazio onde o colocar. Assim o poste auxiliar tem que conter os  $n-1$  discos.

Conclui-se pois que a primeira etapa da resolução do problema 2.1 consiste em transferir  $n-1$  discos (de forma recorrente) do poste inicial para o poste auxiliar utilizando o poste final como poste auxiliar.

É agora possível mover directamente o disco  $n$  do poste inicial para o poste final.

Finalmente é necessário transferir (de forma recorrente) os  $n-1$  discos do poste auxiliar para o poste final, utilizando o poste inicial como auxiliar.

As considerações anteriores podem ser resumidas no algoritmo seguinte:

**Algoritmo 2.1 Torres de Hanoi (Problema 2.1)**

*Hanoi(n, postelnicial, posteAuxiliar, posteFinal)*

*Se  $n = 1$*

*Então MoveDisco(1, postelnicial, posteFinal)*

*senão Hanoi(n-1, postelnicial, posteFinal, posteAuxiliar)*

*MoveDisco(n, postelnicial, posteFinal)*

*Hanoi(n-1, posteAuxiliar, postelnicial, posteFinal).*

Seja  $T(n)$  o número de movimentos necessários para resolver o puzzle de  $n$  discos. Pelo algoritmo anterior,  $T(n)$  é dado pela seguinte fórmula recorrente:

$$T(n) = \begin{cases} 1 & \text{se } n = 1, \\ 2T(n-1) + 1 & \text{se } n > 1. \end{cases}$$

Explicitando a fórmula anterior, conclui-se que o número de movimentos necessários para resolver o problema das Torres de Hanoi com  $n$  discos é  $T(n) = 2^n - 1$ , para  $n \geq 1$ .

Podemos agora verificar que na lenda das Torres de Brahma, por mais rápidos que sejam os sacerdotes a movimentar os discos, digamos ao ritmo de um disco por segundo, ainda teremos que aguardar algum(!) tempo para que o mundo se desvaneça em pó ( $2^{64}-1$  segundos, ou seja, cerca de 584942417 milénios).

### 3. Uma solução Prática

O algoritmo apresentado, apesar da sua simplicidade, não é prático. De facto, qualquer pessoa sentiria dificuldade em resolver o puzzle seguindo o algoritmo recorrente. Vamos explicar como é que este se resolve na prática, começando por analisar quantas vezes se move cada um dos discos:

O disco  $n$  move-se apenas 1 vez (do poste inicial para o poste final). O disco  $n-1$  tem de se mover o dobro das vezes que o disco  $n$  se move, isto é 2 vezes (uma para sair de cima do disco  $n$  e outra para voltar para cima do disco  $n$ ). Analogamente, o disco  $n-2$  tem de se mover o dobro das vezes que o disco  $n-1$  se move, ou seja, 4 (uma para sair de cima do disco  $n-1$  e outra para voltar para cima do disco  $n-1$ , isto repetido tantas vezes quantas o disco  $n-1$  se tem que mover).

Continuando este processo, facilmente se conclui que o menor dos discos, o disco 1, se move o dobro das vezes do disco 2 ou seja  $2^{n-1}$  vezes.

Note-se que estes valores são todos potências de 2, que a soma do número de movimentos dos discos  $n, n-1, \dots, 2$  é  $2+2^2 + \dots + 2^{n-2} = 2^{n-1}-1$ , precisamente menos uma unidade que o número de movimentos do disco mais pequeno e a soma total de movimentos é  $2^{n-1} - 1 + 2^{n-1} = 2^n - 1$  (ver a figura 2).

$$\begin{array}{r} 1 = 2^0 \quad (\text{disco } n) \\ 10 = 2^1 \quad (\text{disco } n-1) \\ 100 = 2^2 \quad (\text{disco } n-2) \\ 1000 = 2^3 \quad (\text{disco } n-3) \\ \dots \\ +1000\dots00000 = 2^{n-1} \quad (\text{disco } 1) \\ 1111\dots 11111 = 2^n - 1 \\ \hline +1 \\ 10000\dots00000 = 2^n \end{array}$$

Figura 2: O número de movimentos da Torre de Hanoi com  $n$  discos, em binário.

Inicialmente temos que mover o disco 1 para a torre destino (no caso  $n$  ímpar) ou para a torre auxiliar (no caso  $n$  par). Em seguida temos que mover alternadamente um dos outros e o disco 1.

De facto, em cada passo da resolução do puzzle das Torres de Hanoi estaremos em uma das duas situações seguintes:

Se no passo anterior o disco movido foi o disco mais pequeno (o disco 1), então não devemos movê-lo novamente porque senão estaríamos a voltar ao passo anterior ou poderíamos ter alcançado o mesmo resultado com um só movimento. Assim, só podemos mover um dos outros discos (o mais pequeno dos que estão no topo dos outros postes) para o poste em que não se encontra o disco 1.

Se no passo anterior o disco movido não foi o mais pequeno, então agora teremos que o mover. Neste caso existem dois postes onde é possível colocar o disco 1. Denominem-se os três postes por  $A$ ,  $B$ ,  $C$  e considere-se que  $A \rightarrow C \rightarrow B \rightarrow A$  é sentido dos ponteiros do relógio. No caso de o número total de discos ser par, o disco 1 desloca-se para o poste seguinte no sentido dos ponteiros do relógio. No caso de  $n$  ser ímpar, o disco 1 desloca-se para o poste seguinte no sentido contrário aos ponteiros do relógio.

## 4. O Algoritmo Iterativo

Qualquer algoritmo recorrente pode ser escrito numa forma não recorrente utilizando o método geral de eliminação de recorrência descrito em vários livros de Algoritmos e Estruturas de Dados [12], [4].

No entanto é preferível tentar tirar partido das propriedades específicas do problema em causa, por forma a

obter um algoritmo não recorrente tão eficiente quanto possível.

De entre os inúmeros autores que têm abordado este problema, como Dijkstra, Hayes ou Walsh, consideramos a solução devida a M. C. Er [3] que utiliza uma curiosa relação entre os movimentos dos discos nas Torres de Hanoi e os números binários (e que resulta directamente da solução apresentada anteriormente):

Um problema que surge de imediato quando se tenta resolver o puzzle das Torres de Hanoi é decidir, em cada iteração, qual dos discos mover. Nesse sentido, designe-se por  $H(n)$  a sequência de movimentos de discos realizada para transferir uma torre de  $n$  discos do poste inicial para o poste final. Tem-se

$$H(n) = H(n-1)nH(n-1)$$

e

$$H(1) = 1.$$

Explicitamente, para mover uma torre de  $n$  discos, movem-se os primeiros  $n-1$  discos (para o poste auxiliar), em seguida move-se o disco  $n$  (do poste inicial para o final) e finalmente movem-se os  $n-1$  discos (para o poste final).

Seja agora  $B(n)$  a sequência de  $2^n - 1$  elementos cuja  $i$ -ésima componente é a representação binária, em  $n$  bits, do número  $i$ . Seja ainda  $R(B(n))$  a sequência de  $2^n - 1$  elementos cujo  $i$ -ésimo elemento é o índice do dígito 1 mais à direita na representação binária de  $i$ . Denotem-se por  $0B(n-1)$  e  $1B(n-1)$  as sequências de  $2^{n-1} - 1$  elementos em que a  $i$ -ésima componente é a  $i$ -ésima componente de  $B(n-1)$  precedida por 0, respectivamente, 1. Finalmente, seja  $Z(n-1)$  a sequência binária constituída por  $n-1$  zeros.

É fácil verificar que (ver figura 3)

$$B(n) = 0B(n-1), 1Z(n-1), 1B(n-1)$$

e que

$$B(1) = 1.$$

$$B(n) \left\{ \begin{array}{l} 0 \ 001 \\ 0 \ 010 \\ 0 \ 011 \\ 0 \ 100 \\ 0 \ 101 \\ 0 \ 110 \\ 0 \ 111 \\ 1 \ 000 \\ 1 \ 001 \\ 1 \ 010 \\ 1 \ 011 \\ 1 \ 100 \\ 1 \ 101 \\ 1 \ 111 \end{array} \right. \begin{array}{l} A(n-1) \\ Z(n-1) \\ A(n-1) \end{array}$$

Figura 3: A sequência  $B(n) = (1, 2, \dots, 2^n - 1)$ , em binário, para  $n=4$ .

A relação entre estas sequências binárias e os movimentos das Torres de Hanoi é estabelecida pelo seguinte teorema:

**Teorema 4.1 (Er)**

$$H(n) = R(B(n)).$$

O teorema anterior permite concluir que na  $i$ -ésima iteração o número do disco a mover é dado pela posição do dígito 1 mais à direita na representação binária de  $i$ .

Resta decidir ainda para que poste é que se move o disco em cada iteração. Considere-se que  $A \rightarrow C \rightarrow B \rightarrow A$  é o sentido dos ponteiros do relógio e que  $A \rightarrow B \rightarrow C \rightarrow A$  é o sentido oposto, onde  $A, B, C$  designam os três postes do puzzle. Er [3] demonstrou que todos os discos ímpares se movimentam num sentido, enquanto os discos pares se movimentam em sentido contrário. O sentido do movimento depende apenas do número total de discos e do poste inicial e final, segundo a regra:

*Se o número de discos  $n$  é ímpar o poste destino é o seguinte ao poste de partida no sentido dos ponteiros do relógio, caso contrário o poste desti-*

*no é o seguinte ao poste de partida no sentido contrário aos ponteiros do relógio.*

Como síntese, apresentamos uma implementação das ideias anteriores na linguagem C:

**Algoritmo 4.1 Torres de Hanoi (versão iterativa binária)**

```
HanoiBinario(int n, char inicial, char final)
{ int limite, i, x, dir;
  for (i=1; i<=n; i++)
    P[i]=inicial;
  dir=(n&1)!=0 ? (final-inicial ==1 ? final-ini-
    cial==2) :
    (final-inicial ==-1 ? final-inicial==1) :
    (final-inicial ==-2);
  limite=(1<<n)-1;
  for(x=1; x<=limite; x++){
    i=0;
    while(!(x>>i&1)) i++;
    printf("mover o disco %d ", ++i);
    printf("de %c ", P[i]);
    printf("para %c\n",
      P[i]='A'+(P[i]-'A'+1+(i&1?dir:1-
        dir))%3);
  }
}
```

## 5. Curiosidades, Variações e Generalizações

### 5.1 Hanoi e Sierpinski

Uma propriedade interessante das Torres de Hanoi surge quando se representam graficamente as configurações possíveis do problema.

Dados  $n$  discos, consideram-se as sequências ordenadas, denominadas configurações,  $(d_1, d_2, \dots, d_n)$  em que  $d_i \in \{A, B, C\}$  indica o poste no qual se encontra o disco  $i$ . A cada uma dessas sequências associa-se o vértice de um grafo. Existe uma aresta entre dois vértices se e só se for

possível passar de uma configuração a outra, sem violar as regras referidas no problema 2.1.

As figuras 4 e 5 ilustram os grafos que se obtêm com 1, 2 e 3 discos, respectivamente.

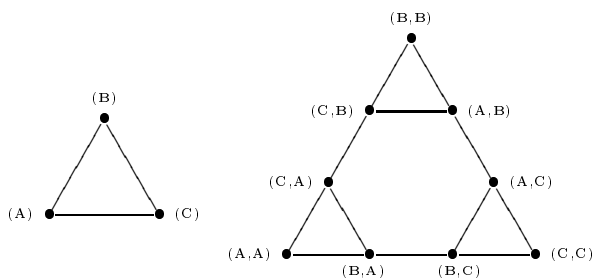


Figura 4: Representação do problema das Torres de Hanoi com 1 e 2 discos.

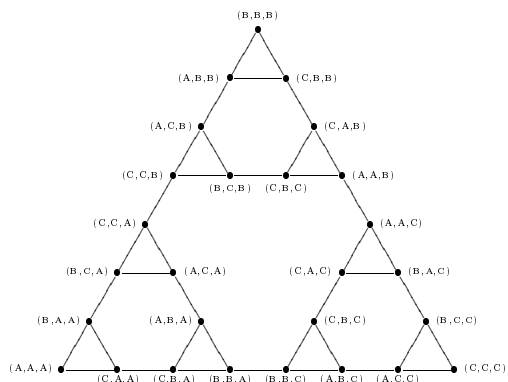


Figura 5: Representação do problema das Torres de Hanoi com 3 discos.

Para um único disco, existem apenas três configurações possíveis: (A), (B) e (C), correspondentes ao poste onde o (único) disco 1 pode estar colocado. Para 2 discos já existem 9 configurações possíveis em que, por exemplo, (B,C) significa que o disco 1 está no poste B e o disco 2 está no poste C. Note-se que o grafo contém apenas configurações possíveis, ou seja, (C,C) corresponde à configuração em que os dois discos estão colocados no poste C e com o disco 1 sobre o disco 2.

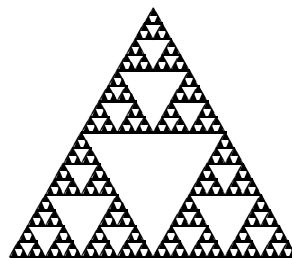


Figura 6: Representação gráfica da curva de Sierpinski.

Como se pode observar nas figuras existe uma semelhança entre os grafos da Torre de Hanoi e a famosa curva de Sierpinski (figura 6). No contexto dos espaços métricos, com a distância de Hausdorff, é possível mostrar que, mantendo constante o tamanho do lado do triângulo exterior, a sequência de grafos de configurações tende para a curva de Sierpinski à medida que  $n$  aumenta.

## 5.2 As Torres de Hanoi Cíclicas

Esta variação do problema das Torres de Hanoi foi inventada em 1981 por Atkinson [1] numa tentativa de elaborar um problema que não apresentasse uma solução iterativa simples.

No entanto não demorou muito para que surgissem publicadas soluções iterativas eficientes, o que não diminui a beleza do problema como se pode constatar pela simplicidade da solução recorrente que passamos a analisar.

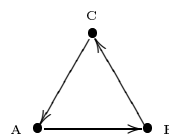


Figura 7: O sentido do movimento dos discos no problema das Torres de Hanoi Cíclica.

Neste puzzle consideram-se os três postes dispostos num triângulo de vértices A, B, C (ver figura 7). Os discos apenas podem ser movidos no sentido contrário aos ponteiros do relógio: de A para B, de B para C ou de C para A.

A solução apresentada por Atkinson é constituída por

dois procedimentos recorrentes, denominados “longo” e “curto” que resolvem, respectivamente, os problemas de deslocar  $n$  discos no sentido dos ponteiros do relógio e deslocar  $n$  discos no sentido inverso:

**Algoritmo 5.1 Torres de Hanoi Cíclicas**

$curto(n);$  Se $n > 0$ então $longo(n-1);$ $moveDisco(n);$ $longo(n-1).$	$longo(n);$  Se $n > 0$ então $longo(n-1);$ $moveDisco(n);$ $curto(n-1);$ $moveDisco(n);$ $longo(n-1);$
--	--

Se denotarmos por  $C(n)$  e  $L(n)$  o número de movimentos realizados, respectivamente, pelos procedimentos *curto* e *longo*, concluímos que

$$\begin{cases} C(n) = 2L(n-1) + 1 & \text{se } n \geq 1 \\ C(0) = 0 \end{cases}$$

e

$$\begin{cases} L(n) = 2L(n-1) + C(n-1) + 2 & \text{se } n \geq 1 \\ L(0) = 0. \end{cases}$$

Demonstra-se que a solução do sistema de equações lineares anteriores é

$$C(n) = \frac{3 + \sqrt{3}}{6} (1 + \sqrt{3})^n + \frac{3 - \sqrt{3}}{6} (1 - \sqrt{3})^n - 1, \quad n \geq 0 \quad (1)$$

$$L(n) = \frac{3 + 2\sqrt{3}}{6} (1 + \sqrt{3})^n + \frac{3 - 2\sqrt{3}}{6} (1 - \sqrt{3})^n - 1, \quad n \geq 0. \quad (2)$$

Segundo Stockmeyer [11], a prova de que estes procedimentos produzem o número mínimo de movimentos não foi apresentada por Atkinson, talvez pensando que tal seria evidente. Stockmeyer demonstra que para além do número mínimo de movimentos, a sequência originada por aqueles procedimentos é única, no sentido de que qual-

quer algoritmo que resolva o problema no número mínimo de movimentos, gera a mesma sequência de movimentos.

### 5.3 Torres de Hanoi em grafos

O puzzle das Torres de Hanoi cíclicas abordado na secção anterior pode ser generalizado para um qualquer grafo fortemente conexo,  $G=(V,E)$ , com  $|V|=3$  vértices.

Os vértices do grafo correspondem aos postes que denotaremos por  $A$ ,  $B$  e  $C$ . Vamos considerar que inicialmente os  $n$  discos se encontram no poste  $A$  e que se pretende transportá-los para o poste  $B$ .

Para além das restrições do problema clássico das Torres de Hanoi, só é possível mover um disco do poste  $i$  para o poste  $j$ , se  $(i,j) \in E$ .

Deixamos ao cuidado do leitor verificar (por indução) que o algoritmo *puzzle3* a seguir apresentado resolve o problema para qualquer grafo nas condições acima enunciadas. Nesse algoritmo a função *auxiliar* serve apenas para seleccionar o poste  $k$ , diferente dos postes  $i$  e  $j$ , por exemplo, o poste  $C$  se os postes  $i$  e  $j$  são  $A$  e  $B$ .

**Algoritmo 5.2 Torres de Hanoi em grafos conexos com 3 vértices**

$puzzle3(i,j,n);$

Se  $n > 0$

então  $k \leftarrow auxiliar(i,j)$

se  $(i,j) \in E$

então  $puzzle3(i,k,n-1);$

$moveDisco(n,i,j);$

$puzzle3(k,j,n-1)$

senão  $puzzle3(i,j,n-1);$

$moveDisco(n,i,k);$

$puzzle3(j,i,n-1);$

$moveDisco(n,k,j);$

$puzzle3(i,j,n-1).$

Seja  $N(i, j, n)$  o número de movimentos realizados pelo algoritmo *puzzle3* ao transferir  $n$  discos do poste  $i$  para o poste  $j$ . Se  $k$  é o poste auxiliar então é fácil verificar que

$$N(i, j, n) = \begin{cases} N(i, k, n-1) + N(k, j, n-1) + 1 & \text{se } (i, j) \in E \\ 2N(i, j, n-1) + N(j, i, n-1) + 2 & \text{se } (i, j) \notin E \end{cases}$$

Pode demonstrar-se que  $N(i, j, n)$  é o número mínimo de movimentos suficientes para transferir a pilha de  $n$  discos do poste  $i$  para o poste  $j$ . Assim o algoritmo *puzzle3* constitui uma solução ótima para o problema em causa.

Vamos passar agora a descrever de forma abreviada como se podem obter explicitamente os valores  $N(i, j, n)$  como função de  $n$ :

Considere-se  $\alpha_{i,j}(n) = N(i, j, n)$ , para  $i, j \in \{A, B, C\}$ ,  $i \neq j$  e  $k$  o poste auxiliar. O problema resume-se a determinar a solução de um sistema de 6 fórmulas recorrentes do tipo

$$\alpha_{i,j}(n) = \begin{cases} \alpha_{i,k}(n-1) + \alpha_{k,j}(n-1) + 1 & \text{se } (i, j) \in E \\ 2\alpha_{i,j}(n-1) + \alpha_{j,i}(n-1) + 2 & \text{se } (i, j) \notin E. \end{cases}$$

Multiplicando as equações do sistema por  $x^n$ , considerando a soma para todos os valores de  $n$ , definindo as funções geradoras

$$f_{i,j}(x) = \sum_{n=0}^{\infty} \alpha_{i,j}(n)x^n,$$

e notando que  $\alpha_{i,j}(0) = 0$ , obtém-se um sistema equivalente de 6 equações da forma:

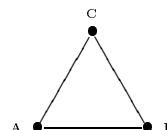
$$\begin{cases} f_{i,j}(x) - xf_{i,k}(x) - xf_{k,j}(x) = \frac{x}{1-x} & \text{se } (i, j) \in E \\ (1-2x)f_{i,j}(x) - xf_{j,i}(x) = \frac{2x}{1-x} & \text{se } (i, j) \notin E \end{cases}$$

Resolvendo o sistema em relação às incógnitas  $f_{i,j}(x)$ ,  $i, j \in \{A, B, C\}$ ,  $i \neq j$ , obtém-se, de forma explícita, as funções geradoras  $f_{i,j}$ . Finalmente, os valores  $N(i, j, n)$  não são mais que os coeficientes da série de Laurent para  $f_{i,j}$ .

É claro que o processo que descrevemos, é moroso e envolve manipulações algébricas bastante complexas. No entanto utilizando software que permita o cálculo simbó-

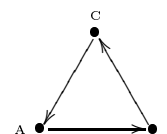
lico, p. ex., o sistema Mathematica [8], é possível obter as seguintes conclusões sobre os grafos fortemente conexos de três vértices, não isomorfos:

### Grafo 1



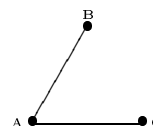
Este é o problema original das Torres de Hanoi e facilmente se conclui que o número de movimentos é, em qualquer caso,  $2^n - 1$ .

### Grafo 2



Este é o grafo do problema das Torres de Hanoi Cíclicas, referido anteriormente. Como seria de esperar, os valores para  $N(i, j, n)$  são dados por (1) e (2), para os dois tipos de movimentos distintos.

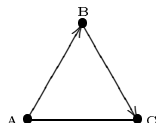
### Grafo 3



Para este grafo conclui-se que existem dois tipos de movimentos:

$$N(A, B, n) = N(B, A, n) = N(A, C, n) = N(C, A, n) = \frac{3^n - 1}{2};$$

$$N(B, C, n) = N(C, B, n) = 3^n - 1.$$

**Grafo 4**

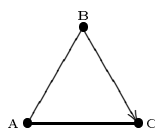
Neste caso temos

$$N(A, B, n) = N(B, C, n) = \frac{-3}{4} - \frac{11-3\sqrt{17}}{8\sqrt{17}} \left( \frac{1-\sqrt{17}}{2} \right)^n + \frac{11+3\sqrt{17}}{8\sqrt{17}} \left( \frac{1+\sqrt{17}}{2} \right)^n;$$

$$N(C, A, n) = \frac{-3}{2} - \frac{4-\sqrt{17}}{2\sqrt{17}} \left( \frac{1-\sqrt{17}}{2} \right)^n + \frac{4+\sqrt{17}}{2\sqrt{17}} \left( \frac{1+\sqrt{17}}{2} \right)^n \text{ para } n \geq 1;$$

$$N(C, B, n) = N(B, A, n) = \frac{-5}{4} - \frac{21-5\sqrt{17}}{8\sqrt{17}} \left( \frac{1-\sqrt{17}}{2} \right)^n + \frac{21+5\sqrt{17}}{8\sqrt{17}} \left( \frac{1+\sqrt{17}}{2} \right)^n;$$

$$N(A, C, n) = \frac{-1}{2} - \frac{5-\sqrt{17}}{4\sqrt{17}} \left( \frac{1-\sqrt{17}}{2} \right)^n + \frac{5+\sqrt{17}}{4\sqrt{17}} \left( \frac{1+\sqrt{17}}{2} \right)^n.$$

**Grafo 5**

Aqui as funções geradoras de  $f_{i,j}(x)$  são funções racionais de denominador  $2x^3 - 4x^2 - x + 1$ . Os coeficientes  $\alpha_{i,j}(n)$  são expressões complexas, para as quais não se conhece uma representação simples. Aqui existem quatro tipos de movimentos distintos, sendo  $N(A, B, n) = N(C, A, n)$  e  $N(A, C, n) = N(B, A, n)$ . Simulações efectuadas, sugerem que a

ordem de complexidade da solução é, em qualquer caso, aproximadamente  $(2.14)^n$ .

Na tabela 1 pode consultar os valores de  $N(i, j, n)$  para diversos valores de  $n$ .

$n$	$N(A, B, n)$	$N(A, C, n)$	$N(B, C, n)$	$N(C, B, n)$
1	1	1	1	2
2	4	3	3	7
3	11	8	7	19
4	28	19	17	47
5	67	46	39	113
6	160	107	93	267
7	375	254	215	629
8	884	591	509	1475
9	2067	1394	1183	3461
10	4856	3251	2789	8107

Tabela 1: O número de movimentos realizados pelo algoritmo *puzzle3* com o grafo 5.

**5.4 Torres de Hanoi com 4 postes**

A primeira versão das Torres de Hanoi com 4 postes, surgiu no livro de Henry Dudeney, *The Canterbury Puzzles* [2], sob a forma de um desafio que consistia em mover uma pilha de queijos de vários tamanhos colocados na primeira de quatro mesas, para uma das outras, sem colocar um queijo maior sobre um menor.

O puzzle foi publicado posteriormente na Revista *American Mathematical Monthly* [9], em 1939, generalizado para um número arbitrário de postes, sob a designação de *Problema 3918*.

Várias propostas de resolução foram apresentadas, sem contudo ter sido demonstrada a sua optimalidade.

Um dos algoritmos mais conhecidos, baseia-se num parâmetro  $i$ , com  $1 \leq i \leq n$ , e consiste nos seguintes passos:

1. Transferir a pilha dos  $n - i$  menores discos do primeiro poste para um poste auxiliar, usando os quatro postes no processo;
2. Transferir a pilha dos restantes  $i$  discos do primeiro pos-



te para o poste destino, utilizando o algoritmo original das Torres de Hanoi com 3 postes (ignorando o poste que contém os  $n-i$  discos mais pequenos);

3. Transferir os  $n-i$  discos mais pequenos do poste auxiliar para o poste final, utilizando novamente os quatro postes no processo.

Demonstra-se que se  $n$  é um número triangular<sup>1</sup>  $t_k$ , então a escolha ótima para  $i$  é  $i=k$ . Mais, se  $t_{k-1} < n < t_k$  então tanto  $k-1$  como  $k$  são valores ótimos para o parâmetro  $i$ .

Note-se que está demonstrado apenas que entre todos os valores possíveis para o parâmetro  $i$ , os referidos anteriormente são os que minimizam o número de movimentos para aquele algoritmo. Não se tem conhecimento se o algoritmo ótimo tem a forma apresentada, o que constitui a chamada conjectura de Frame-Stewart, em homenagem aos primeiros autores a apresentarem aquele algoritmo.

Quanto ao número de movimentos realizados por este algoritmo, Stockmeyer [10], provou que é da ordem de  $\sqrt{n} \cdot 2^{\sqrt{2n}}$ .

Por fim, têm aparecido várias propostas de resolução da versão cíclica do problema das Torres de Hanoi com 4 postes, mas todos os algoritmos apresentados falham a optimalidade, sendo este mais um problema em aberto.

## Referências

- [1] M. D. Atkison. The Cyclic Towers of Hanoi. *Information Processing Letters*, (13): 118-119, 1981.
- [2] H. E. Dudeney. *The Canterbury Puzzles*. Thomas Nelson & Sons, London, 1907.
- [3] M. C. Er. Performance evaluations of recursive as iterative algorithms for the towers of Hanoi problem. *Computing*, (37): 93-102, 1986.

- [4] Helman and R. Veroff, *Intermediate Problem Solving and Data Structures: Walls and Mirrors*, Benjamin-Cummings, 1986.
- [5] R. D. Hofstadter. Metamagical themas. *Scientific American*, 2(248): 16-22, Março 1983.
- [6] É. Lucas. Nouveaux jeux scientifiques. *La Nature*. 17:301-303, 1889.
- [7] É. Lucas. *Récréations mathématiques*, 1893. Reeditado diversas vezes por Albert Blanchard, Paris.
- [8] Mathematica. Wolfram Research, <http://www.wolfram.com>.
- [9] B. M. Stewart. Advanced Problem 3918. *American Mathematical Monthly*, 1939.
- [10] P. K. Stockmeyer. Variations on the four-post tower of Hanoi puzzle. *Congressus Numerantium*, pages 3-12, 1994.
- [11] P. K. Stockmeyer. The average distance between nodes in the cyclic towers of Hanoi digraph. *Graph Theory, Combinatorics, Algorithms, and Applications*, 1996.
- [12] N. Wirth. *Algorithms + Data Structures = Programs*. Prentice-Hall, 1976.

### Gödel em Princeton

Alguma coisa pensa em  
si própria em mim.  
Em algum tempo ou em algum lugar  
alguma coisa é real, pensando.

Às vezes quase lhe toco  
quando não me perturbam os meus pensamentos.  
E talvez quando faço  
sem dar por isso os gestos de todos os dias  
talvez então esteja muito perto sem o saber.

E alguém me leve pela mão por uma realidade  
feita da minha vida e de coisas reais  
a que pertencemos eu e o que pensa.

Manuel António Pina,  
in "Poesia Reunida", Assírio & Alvim, 2001.  
(publicação gentilmente autorizada pelo autor)

<sup>1</sup>  $t_k = \binom{k}{2} = \frac{k(k-1)}{2}$