Análise de Algoritmos: Torres de Hanói

SCC201/501/601 - Introdução à Ciência de Computação II

Prof. Moacir A. Ponti

www.icmc.usp.br/~moacir

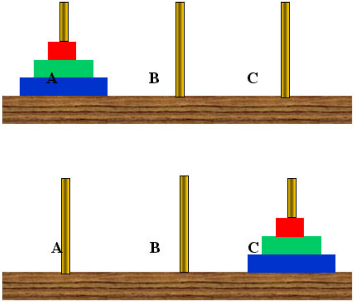
Instituto de Ciências Matemáticas e de Computação – USP

2020/2

Sumário

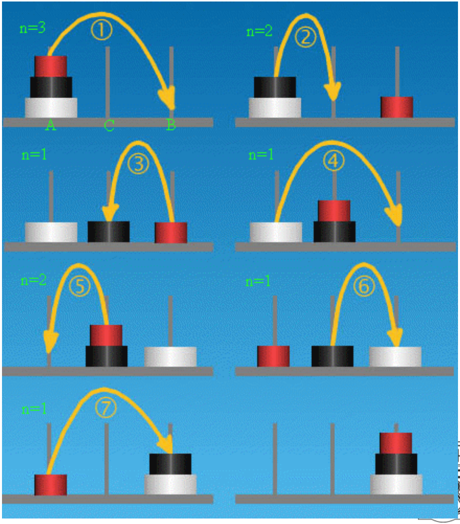
Torres de Hanói

- Problema que consiste em três postes e um número de discos de diferentes tamanhos que podem deslizar pelos postes.
- É preciso mover os discos de um poste a outro seguindo as seguintes regras: a) mover um disco de cada vez e b) um disco maior não pode ficar sobre um disco menor.



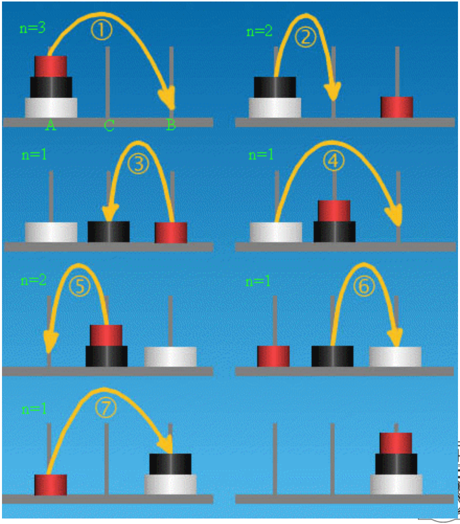
Torres de Hanói

- Tentar resolver esse problema é um bom exercício de como pensar recursivamente
- É preciso entender o caso base e como reduzir o problema a uma instância menor.



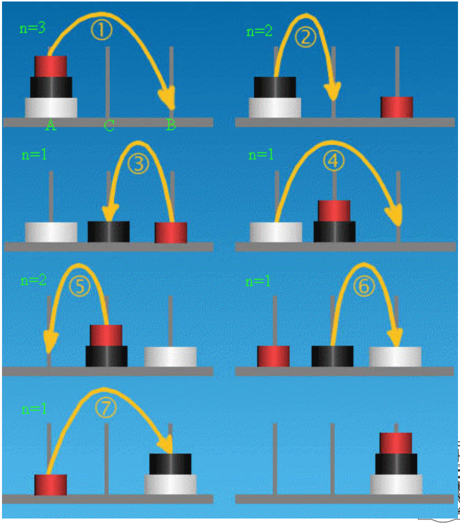
Torres de Hanói

- Tentar resolver esse problema é um bom exercício de como pensar recursivamente
- É preciso entender o caso base e como reduzir o problema a uma instância menor.
- A estratégia básica é:
  1. Mover $n - 1$ discos do poste origem para o intermediário

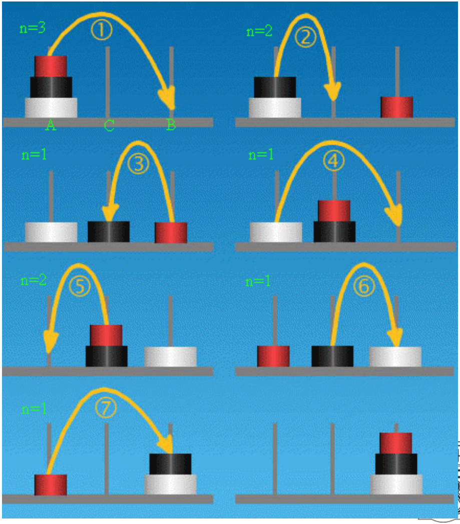


Torres de Hanói

- Tentar resolver esse problema é um bom exercício de como pensar recursivamente
- É preciso entender o caso base e como reduzir o problema a uma instância menor.
- A estratégia básica é:
    1. Mover $n - 1$ discos do poste origem para o intermediário
    2. Mover 1 disco (disco base) do poste origem para o destino



- Tentar resolver esse problema é um bom exercício de como pensar recursivamente
- É preciso entender o caso base e como reduzir o problema a uma instância menor.
- A estratégia básica é:
  1. Mover $n-1$ discos do poste origem para o intermediário
  2. Mover 1 disco (disco base) do poste origem para o destino
  3. Mover $n-1$ discos do poste intermediário para o destino



Torres de Hanói

```
  void Hanoi(int tam, char ori, char des, char interm) {
1     if (tam == 1)
2         printf("Mova disco de %c para %c\n", ori, des);
    else {
3         Hanoi(tam-1, ori, interm, des);
4         Hanoi(1, ori, des, interm);
5         Hanoi(tam-1, interm, des, ori);
    }
  }
```

- Qual é a ordem de crescimento desse algoritmo? (considere $T(1) = 1$)

Torres de Hanói

```
 void Hanoi(int tam, char ori, char des, char interm) {
1   if (tam == 1)
2      printf("Mova disco de %c para %c\n", ori, des);
   else {
3      Hanoi(tam-1, ori, interm, des);
4      Hanoi(1, ori, des, interm);
5      Hanoi(tam-1, interm, des, ori);
   }
 }
```

- Qual é a ordem de crescimento desse algoritmo? (considere $T(1) = 1$)
- Para encontrar uma fórmula temos: uma movimentação de 1 disco e duas movimentações de $n-1$ discos.

Torres de Hanói

```
 void Hanoi(int tam, char ori, char des, char interm) {
1   if (tam == 1)
2      printf("Mova disco de %c para %c\n", ori, des);
   else {
3      Hanoi(tam-1, ori, interm, des);
4      Hanoi(1, ori, des, interm);
5      Hanoi(tam-1, interm, des, ori);
   }
 }
```

- Qual é a ordem de crescimento desse algoritmo? (considere $T(1) = 1$)
- Para encontrar uma fórmula temos: uma movimentação de 1 disco e duas movimentações de $n-1$ discos.
- $T(n) = T(n-1) + T(1) + T(n-1)$

Torres de Hanói

Fórmula básica

$$
\begin{aligned}
T(n) &= T(n-1) + T(1) + T(n-1) \\
&= 2 \cdot T(n-1) + T(1) \\
&= 2 \cdot T(n-1) + 1
\end{aligned}
$$

expandindo...

$$
\begin{aligned}
T(n) &= 2 \cdot T(n-1) + 1 \\
T(n) &= 2 \cdot (2 \cdot T(n-2) + 1) + 1 \\
&= 2^2 \cdot T(n-2) + 2^1 + 2^0 \\
&\dots
\end{aligned}
$$

expandindo...

$$
\begin{aligned}
T(n) &= 2 \cdot T(n-1) + 1 \\
T(n) &= 2 \cdot (2 \cdot T(n-2) + 1) + 1 \\
&= 2^2 \cdot T(n-2) + 2^1 + 2^0 \\
&\ldots \\
&= 2 \cdot (2 \cdot (2 \cdot T(n-3) + 1) + 1) + 1 \\
&= 2^3 \cdot T(n-3) + 2^2 + 2^1 + 2^0 \\
&\ldots
\end{aligned}
$$

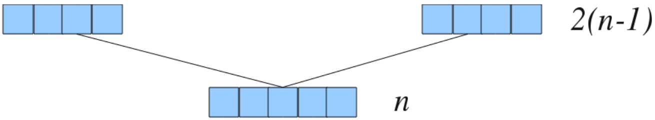
expandindo...

$$
\begin{aligned}
T(n) &= 2 \cdot T(n-1) + 1 \\
T(n) &= 2 \cdot (2 \cdot T(n-2) + 1) + 1 \\
&= 2^2 \cdot T(n-2) + 2^1 + 2^0 \\
&\ldots \\
&= 2 \cdot (2 \cdot (2 \cdot T(n-3) + 1) + 1) + 1 \\
&= 2^3 \cdot T(n-3) + 2^2 + 2^1 + 2^0 \\
&\ldots \\
T(n) &= 2^k \cdot T(n-k) + 2^{(k-1)} + 2^{(k-2)} + \cdots + 2^0 \\
T(n) &= 2^k \cdot T(n-k) + \sum_{i=0}^{k-1} 2^i
\end{aligned}
$$

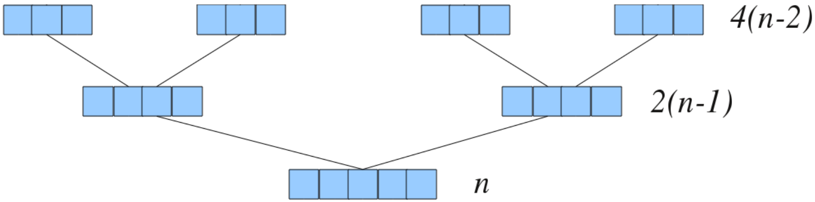
Torres de Hanói

n

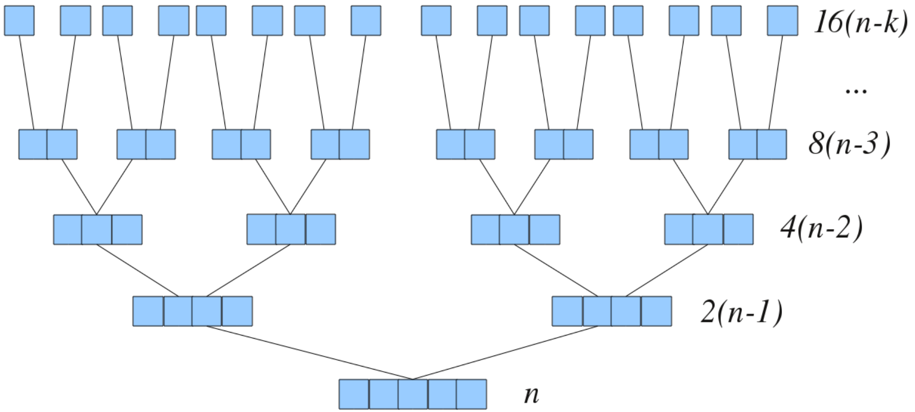
Torres de Hanói



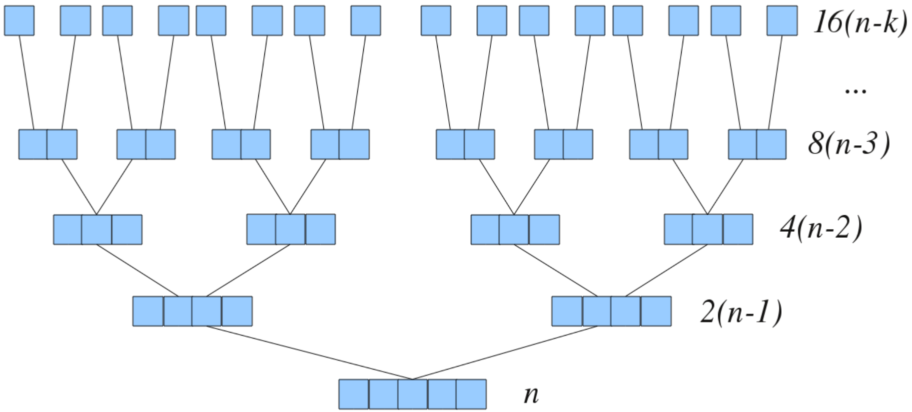
Torres de Hanói



Torres de Hanói



Torres de Hanói



Torres de Hanói

- Para alcançar o caso base, $T(n-k=1)$, é preciso que $k=n-1$:

$$T(n) = 2^{n-1} \cdot T(1) + \sum_{i=0}^{n-1-1} 2^i$$
$$= 2^{n-1} \cdot 1 + \sum_{i=0}^{n-2} 2^i$$

Por soma de Progressão Geométrica, e simplificando:

$$T(n) = 2^{n-1} + 2^{n-1} - 1 = 2^n - 1$$

- Para alcançar o caso base, $T(n-k=1)$, é preciso que $k=n-1$:

$$T(n) = 2^{n-1} \cdot T(1) + \sum_{i=0}^{n-1-1} 2^i$$
$$= 2^{n-1} \cdot 1 + \sum_{i=0}^{n-2} 2^i$$

Por soma de Progressão Geométrica, e simplificando:

$$T(n) = 2^{n-1} + 2^{n-1} - 1 = 2^n - 1$$

- A complexidade do problema é de ordem exponencial, mais especificamente $O(2^n)$.
- A cada passo 1 unidade é resolvida e o problema é dividido em **2** partes menores (de tamanho $n-1$)

Torres de Hanói

- Problema inventado por Édouard Lucas em 1883, com base em uma lenda (inventada por ele ou que o inspirou?).

Torres de Hanói

- Problema inventado por Édouard Lucas em 1883, com base em uma lenda (inventada por ele ou que o inspirou?).
- O criador do universo, no início dos tempos criou em Hanoi uma grande sala com três postes. Num dos postes colocou 64 discos dourados de tamanhos diferentes, do maior para o menor.
- Os sacerdotes de Hanói, criados na mesma época, de acordo com a lenda, realizam movimentos com os discos de um poste para outro seguindo as duas regras do problema.

Torres de Hanói

- Problema inventado por Édouard Lucas em 1883, com base em uma lenda (inventada por ele ou que o inspirou?).
- O criador do universo, no início dos tempos criou em Hanoi uma grande sala com três postes. Num dos postes colocou 64 discos dourados de tamanhos diferentes, do maior para o menor.
- Os sacerdotes de Hanói, criados na mesma época, de acordo com a lenda, realizam movimentos com os discos de um poste para outro seguindo as duas regras do problema.
- Segundo a estória, quando o último movimento do quebra-cabeças for feito, o mundo chegara ao fim.

Bibliografia

- ZIVIANI, N. **Projeto de algoritmos**: com implementações em Pascal e C. (seção 1.4). 2.ed. Thomson, 2004.
- CORMEN, T.H. et al. **Algoritmos: Teoria e Prática** (Capítulo 4). Campus. 2002.
- FEOFILOFF, P. **Recorrências**. Disponível em: `http://www.ime.usp.br/~pf/analise_de_algoritmos/aulas/recorrencias.html`.