MAC0122 Princípios de Desenvolvimento de Algoritmos

Edição 2012

AULA 1

Administração

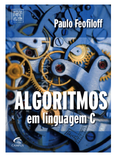
Página da disciplina: aulas, cadastro, eps ...

`http://paca.ime.usp.br/`

Chave de inscrição: grauna

Exercício programa 1: disponível na página

Livros



Nossa referência básica é o livro
PF = Paulo Feofiloff,
Algoritmos em linguagem C,

Este livro é baseado no material do sítio

Projeto de Algoritmos em C.

Outros livros são
S = Robert Sedgewick,
Algorithms in C, vol. 1

CLRS = Cormen-Leiserson-Rivest-Stein,
Introductions to Algorithms

MAC0122

MAC0122 Princípios de Desenvolvimento de Algoritmos é uma disciplina introdutória em:

- projeto, correção e eficiência de algoritmos e
- estruturas de dados

MAC0122

MAC0122 combina técnicas de

- programação
- correção de algoritmos (relações invariantes)
- análise da eficiência de algoritmos e
- estruturas de dados elementares

que nasceram de aplicações cotidianas em ciência da computação.

Pré-requisitos

O pré-requisito oficial de MAC0122 é

- MAC0110 Introdução à Computação.

Costuma ser conveniente ter cursado

- MAT0138 Álgebra I para Computação

Principais tópicos

Alguns dos tópicos de MAC0122 são:

- recursão;
- busca em um vetor;
- busca (binária) em vetor ordenado;
- listas encadeadas;
- listas lineares: filas e pilhas;
- algoritmos de enumeração;
- busca de palavras em um texto;
- algoritmos de ordenação: bubblesort, heapsort, mergesort,...; e

Tudo isso regado a muita análise de eficiência de algoritmos e invariantes.

Localização

MAC0122 é um primeiro passo para

- MAC0323 Estruturas de Dados
- MAC0328 Algoritmos de Grafos
- MAC0338 Análise de Algoritmos

Entretanto, várias outras disciplina se apoiam em MAC0122.

Pausa para nossos comerciais

- Encontro do BCC: 06, 07, 08, 09 e 11 de agosto

  Bio-Informática, Ferramentas Livres para Teste de Invasão, Alan Turing e suas principais contribuições e muito mais.

  http://www.ime.usp.br/~encontrobcc/2012/
- XVI Maratona de Programação: 18 de agosto

  http://www.ime.usp.br/~cef/XVImaratona/
- Página do BCC:

  http://bcc.ime.usp.br/

Recursão

PF 2.1, 2.2, 2.3   S 5.1

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

Recursão

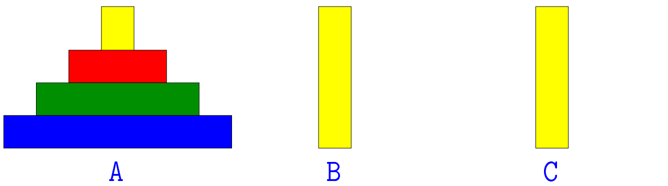
"To understand recursion, we must first understand recursion."

–folclore

"Para fazer uma função recursivo é preciso ter fé."

–Siang Wu Song

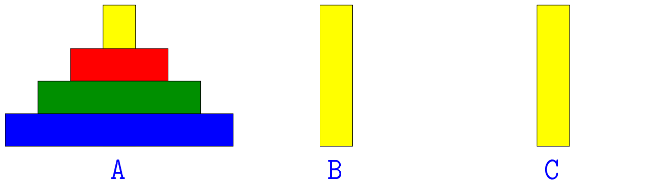
Torres de Hanoi



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

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

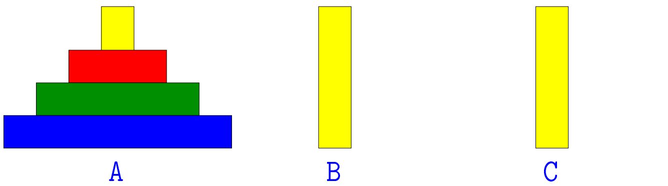
Torres de Hanoi



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

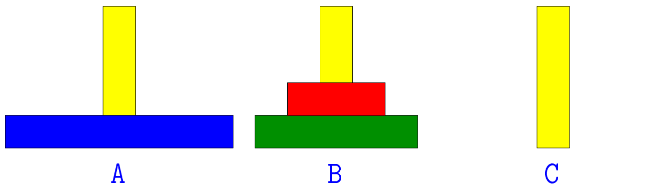
Como resolver Hanoi(n,A,B,C)?

Idéia



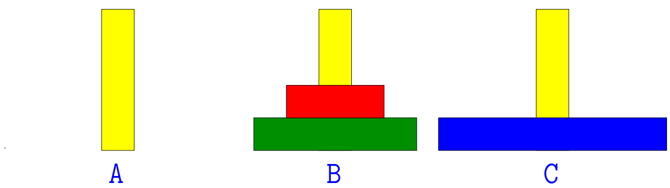
Posso não saber qual o primeiro movimento, mas é fácil saber qual é o **movimento do meio**.

Idéia



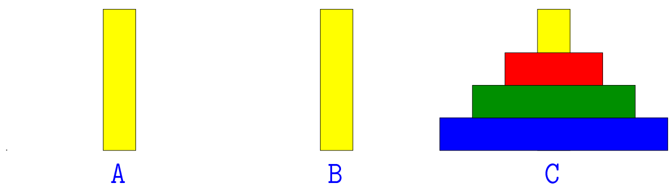
Posso não saber qual o primeiro movimento, mas é fácil saber qual é o **movimento do meio**.

Idéia



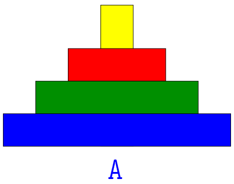
Posso não saber qual o primeiro movimento, mas é fácil saber qual é o **movimento do meio**.

Idéia



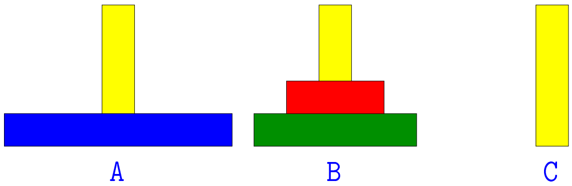
Posso não saber qual o primeiro movimento, mas é fácil saber qual é o **movimento do meio**.

Solução



Para resolver Hanoi(n,A,B,C) basta:

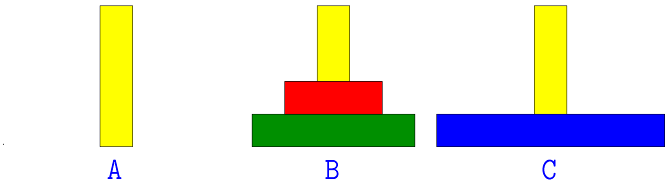
Solução



Para resolver Hanoi(n,A,B,C) basta:

1. resolver Hanoi(<u>n-1</u>,A,C,B)

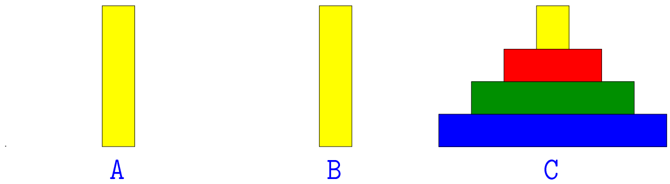
Solução



Para resolver Hanoi(n,A,B,C) basta:

1. resolver Hanoi(<u>n-1</u>,A,C,B)
2. mover o disco n de A para C

Solução



Para resolver Hanoi(n,A,B,C) basta:

1. resolver Hanoi(<u>n-1</u>,A,C,B)
2. mover o disco n de A para C
3. resolver Hanoi(<u>n-1</u>,B,A,C)

Solução

Para resolver Hanoi(n,A,B,C) basta:

1. resolver Hanoi(<u>n-1</u>,A,C,B)
2. mover o disco n de A para C
3. resolver Hanoi(<u>n-1</u>,B,A,C)

E dai?

Solução

Para resolver Hanoi(n,A,B,C) basta:

1. resolver Hanoi(<u>n-1</u>,A,C,B)
2. mover o disco n de A para C
3. resolver Hanoi(<u>n-1</u>,B,A,C)

E dai?

Reduzimos o problema <u>com n discos</u> para 2 problema <u>com n-1 disco</u>!

Solução

Para resolver Hanoi(n,A,B,C) basta:

1. resolver Hanoi(n-1,A,C,B)
2. mover o disco n de A para C
3. resolver Hanoi(n-1,B,A,C)

E dai?

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

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

Hanoi(**0**,. . . ,. . . ,. . .)

Função que resolve o problema

```
void
hanoi(int n, char origem, char auxiliar, char destino)
{
  if (n > 0)
    {
       hanoi(n-1, origem, destino, auxiliar);
       print("mova o disco %d de %c para %c.\n",
              n, origem, destino);
       hanoi(n-1, auxiliar, origem, destino);
    }
}
```

Chamada: `hanoi(n,'A','B','C');`

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.
```

Hanoi(7,A,B,C)

```
1: mova o disco 1 do pino A para 44: mova o disco 3 do pino B para 87: mova o disco 1 do pino C para o pin
2: mova o disco 2 do pino A para 45: mova o disco 1 do pino C para 88: mova o disco 4 do pino C para o pin
3: mova o disco 1 do pino C para 46: mova o disco 2 do pino C para 89: mova o disco 1 do pino B para o pin
4: mova o disco 3 do pino A para 47: mova o disco 1 do pino B para 90: mova o disco 2 do pino B para o pin
5: mova o disco 1 do pino B para 48: mova o disco 5 do pino C para 91: mova o disco 1 do pino A para o pin
6: mova o disco 2 do pino B para 49: mova o disco 1 do pino A para 92: mova o disco 3 do pino B para o pin
7: mova o disco 1 do pino A para 50: mova o disco 2 do pino A para 93: mova o disco 1 do pino C para o pin
8: mova o disco 4 do pino A para 51: mova o disco 1 do pino C para 94: mova o disco 2 do pino C para o pin
9: mova o disco 1 do pino C para 52: mova o disco 3 do pino A para 95: mova o disco 1 do pino B para o pin
10: mova o disco 2 do pino C para 53: mova o disco 1 do pino B para 96: mova o disco 6 do pino B para o pin
11: mova o disco 1 do pino B para 54: mova o disco 2 do pino B para 97: mova o disco 1 do pino A para o pin
12: mova o disco 3 do pino C para 55: mova o disco 1 do pino A para 98: mova o disco 2 do pino A para o pin
13: mova o disco 1 do pino A para 56: mova o disco 4 do pino A para 99: mova o disco 1 do pino C para o pin
14: mova o disco 2 do pino A para 57: mova o disco 1 do pino C para 100: mova o disco 3 do pino A para o pi
15: mova o disco 1 do pino C para 58: mova o disco 2 do pino C para 101: mova o disco 1 do pino B para o pi
16: mova o disco 5 do pino A para 59: mova o disco 1 do pino B para 102: mova o disco 2 do pino B para o pi
17: mova o disco 1 do pino B para 60: mova o disco 3 do pino C para 103: mova o disco 1 do pino A para o pi
18: mova o disco 2 do pino B para 61: mova o disco 1 do pino A para 104: mova o disco 4 do pino A para o pi
19: mova o disco 1 do pino A para 62: mova o disco 2 do pino A para 105: mova o disco 1 do pino C para o pi
20: mova o disco 3 do pino B para 63: mova o disco 1 do pino C para 106: mova o disco 2 do pino C para o pi
21: mova o disco 1 do pino C para 64: mova o disco 7 do pino A para 107: mova o disco 1 do pino B para o pi
22: mova o disco 2 do pino C para 65: mova o disco 1 do pino B para 108: mova o disco 3 do pino C para o pi
23: mova o disco 1 do pino B para 66: mova o disco 2 do pino B para 109: mova o disco 1 do pino A para o pi
24: mova o disco 4 do pino B para 67: mova o disco 1 do pino A para 110: mova o disco 2 do pino A para o pi
25: mova o disco 1 do pino A para 68: mova o disco 3 do pino B para 111: mova o disco 1 do pino C para o pi
26: mova o disco 2 do pino A para 69: mova o disco 1 do pino C para 112: mova o disco 5 do pino A para o pi
27: mova o disco 1 do pino C para 70: mova o disco 2 do pino C para 113: mova o disco 1 do pino B para o pi
28: mova o disco 3 do pino A para 71: mova o disco 1 do pino B para 114: mova o disco 2 do pino B para o pi
29: mova o disco 1 do pino B para 72: mova o disco 4 do pino B para 115: mova o disco 1 do pino A para o pi
30: mova o disco 2 do pino B para 73: mova o disco 1 do pino A para 116: mova o disco 3 do pino B para o pi
31: mova o disco 1 do pino A para 74: mova o disco 2 do pino A para 117: mova o disco 1 do pino C para o pi
32: mova o disco 6 do pino A para 75: mova o disco 1 do pino C para 118: mova o disco 2 do pino C para o pi
33: mova o disco 1 do pino C para 76: mova o disco 3 do pino A para 119: mova o disco 1 do pino B para o pi
34: mova o disco 2 do pino C para 77: mova o disco 1 do pino B para 120: mova o disco 4 do pino B para o pi
35: mova o disco 1 do pino B para 78: mova o disco 2 do pino B para 121: mova o disco 1 do pino A para o pi
```

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.

Fatorial recursivo

$$n! = \begin{cases} 1, & \text{quando } n = 0, \\ n \times (n-1)!, & \text{quando } n > 0. \end{cases}$$

```
long
fatorial(long n)
{
   if (n == 0) return 1;
   return n * fatorial(n-1);
}
```

fatorial(10)

```
fatorial(10)
  fatorial(9)
    fatorial(8)
      fatorial(7)
        fatorial(6)
          fatorial(5)
            fatorial(4)
              fatorial(3)
                fatorial(2)
                  fatorial(1)
                    fatorial(0)
fatorial de 10 e' 3628800.
```

Fatorial iterativo

```
long fatorial (int n)
{
  int  i;
  long ifat;

  ifat = 1;
  for (i = 1; i <= n; i++)
    /* neste ponto vale que ifat = (i-1)! */
    ifat = ifat * i;

  return ifat;
}
```