Estruturas de Dados

Victor Machado da Silva, MSc
victor.silva@professores.ibmec.edu.br

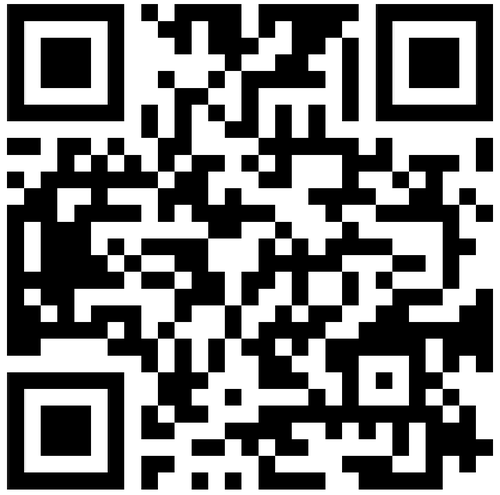
ibmec

Índice

Apresentação do curso

Apresentação do curso

- Contato: victor.silva@professores.ibmec.edu.br
- Grupo no Whatsapp: https://chat.whatsapp.com/IqnSl5PTiVBI1WNwMpXxSS
- Material: www.victor0machado.github.io



Apresentação do curso

- Avaliação
  - Proporção
    - AC (20%): atividades em sala
    - AP1 (40%): projeto + prova objetiva (sem consulta)
    - AP2 (40%): projeto + prova objetiva (sem consulta)
  - Detalhes das entregas
    - Atividades da AC individuais ou em dupla
    - Projetos da AP1 e AP2 em grupos, mínimo 2 e máximo 3 pessoas

Algoritmos

Introdução

Um algoritmo é um processo sistemático para a resolução de um problema. O desenvolvimento de algoritmos é particularmente importante para problemas a serem solucionados em um computador, pela própria natureza do instrumento utilizado.

Um algoritmo computa uma saída, o resultado do problema, a partir de uma entrada, as informações inicialmente conhecidas e que permitem encontrar a solução do problema. Durante o processo de computação o algoritmo manipula dados, gerados a partir da sua entrada.

O estudo de estruturas de dados não pode ser desvinculado de seus aspectos algorítmicos. A escolha correta da estrutura adequada a cada caso depende diretamente do conhecimento de algoritmos para manipular a estrutura de maneira eficiente.

Apresentação dos algoritmos

As convenções seguintes serão utilizadas com respeito à linguagem:

- O início e o final de cada bloco são determinados por indentação, isto é, pela posição da margem esquerda. Se uma certa linha do algoritmo inicia um bloco, ele se estende até a última linha seguinte, cuja margem esquerda se localiza mais à direita do que a primeira do bloco;
- A declaração de atribuição é indicada pelo símbolo :=;
- As declarações seguintes são empregadas com significado semelhante ao usual:

```
se... então
se... então... senão
enquanto... faça
para... faça
pare
```

- Variáveis simples, vetores, matrizes e registro são considerados como tradicionalmente em linguagens de programação. Os elementos de vetores e matrizes são identificados por índices entre colchetes.

```
para i := 1, ..., |__n/2__|
    temp := S[i]
    S[i] := S[n - i + 1]
    S[n - i + 1] := temp
```

Aplicações

- Escreva os algoritmos, em pseudocódigo, para os problemas abaixo:
  - https://br.spoj.com/problems/TOMADA13/
  - https://br.spoj.com/problems/METEORO/
  - https://br.spoj.com/problems/JDESAF12/
  - https://br.spoj.com/problems/CARTAS14/
  - https://br.spoj.com/problems/ENCOTEL/

Recursividade

Um tipo especial de procedimento será utilizado, algumas vezes, ao longo do curso. É aquele que contém, em sua descrição, uma ou mais chamadas a si mesmo. Um procedimento dessa natureza é denominado **recursivo**.

Naturalmente, todo procedimento, recursivo ou não, deve possuir pelo menos uma chamada proveniente de um local exterior a ele. Essa chamada é denominada **externa**. Um procedimento não recursivo é, pois, aquele em que todas as chamadas são externas.

O exemplo clássico mais simples de recursividade é o cálculo do fatorial de um inteiro $n \geq 0$:

```
função fat(i)
    fat(i) := se i <= 1 então 1 senão i * fat(i - 1)
```

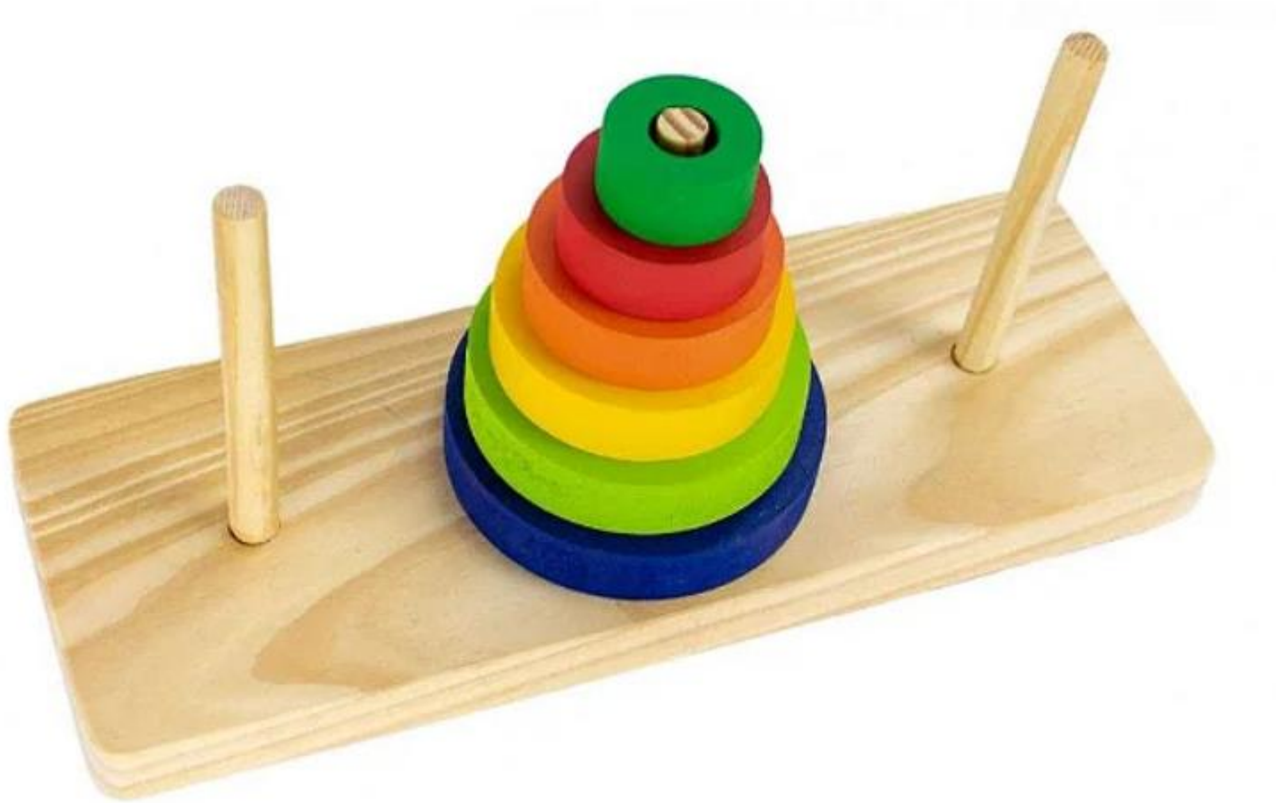
```
fat[0] := 1
para j := 1, ..., n faça
    fat[j] := j * fat[j - 1]
```

Aplicações

- Escreva os algoritmos, em pseudocódigo, para os problemas abaixo:
  - https://br.spoj.com/problems/F91/
  - https://br.spoj.com/problems/RUMO9S/
  - https://br.spoj.com/problems/PARIDADE/

Recursividade

Um exemplo conhecido, onde a solução recursiva é comum e extremamente mais simples que a solução não-recursiva, é o do Problema da Torre de Hanói.



Recursividade

A solução do problema é descrita a seguir. Para $n > 1$, o pino-trabalho deve ser utilizado como área de armazenamento temporário. O raciocínio utilizado para resolver o problema é semelhante ao de uma prova matemática por indução. Suponha que se saiba como resolver o problema até $n - 1$ discos, $n > 1$, de forma recursiva. A extensão para n discos pode ser obtida pela realização dos seguintes passos:

- Resolver o problema da Torre de Hanói para os $n - 1$ discos do topo do pino-origem A, supondo que o pino-destino seja C e o trabalho seja B;
- Mover o n-ésimo pino (maior de todos) de A para B;
- Resolver o problema da Torre de Hanói para os $n - 1$ discos localizados no pino C, suposto origem, considerando os pinos A e B como trabalho e destino, respectivamente.

```
procedimento hanoi(n, A, B, C)
se n > 0 então
    hanoi(n - 1, A, C, B)
    mover o disco do topo de A para B
    hanoi(n - 1, C, B, A)
```

IBMEC.BR

/IBMEC

IBMEC

@IBMEC_OFICIAL

@IBMEC

Ibmec