Estrutura de Dados

Jayme Luiz Szwarcfiter

Fabio Protti

cederj

Aula 1: Introdução

- Apresentação dos algoritmos
- Algoritmo de inversão de uma seqüência
- Algoritmo de cálculo de fatorial

Composição do Curso

- 4 módulos, total de 36 aulas
- No decorrer das aulas:
  - Exercícios a resolver
  - Exercícios finais
- Provas

Módulo 1: Conceitos Básicos e Listas Lineares

- Aula 1: Introdução
- Aula 2: Recursividade
- Aula 3: Complexidade de Algoritmos
- Aula 4: Notação O
- Aula 5: Listas Lineares
- Aula 6: Manipulação de Listas Lineares
- Aula 7: Caso Médio da Busca Linear
- Aula 8: Busca Binária
- Aula 9: Ordenação de Listas Lineares

Módulo 2: Pilhas, Filas e Alocação Encadeada

- Aula 10: Pilhas
- Aula 11: Filas
- Aula 12: Lista Encadeada
- Aula 13: Listas Simplesmente Encadeadas
- Aula 14: Manipulação de Listas Simplesmente Encadeadas
- Aula 15: Listas Duplamente Encadeadas
- Aula 16: Manipulação de Listas Duplamente Encadeadas

Módulo 3: Árvores e Listas Encadeadas

- Aula 17: Conceitos de Árvores e Árvores Binárias
- Aula 18: Propriedades de Árvores Binárias
- Aula 19: Percursos em Árvores Binárias
- Aula 20: Árvores Binárias de Busca
- Aula 21: Freqüências de Acesso Diferenciadas
- Aula 22: Algoritmo de Obtenção da Árvore Ótima
- Aula 23: Árvores Balanceadas
- Aula 24: Inclusão em Árvores AVL
- Aula 25: Árvores Graduadas e Rubro-Negras
- Aula 26: Árvores B
- Aula 27: Manipulação de Árvores B

Módulo 4: Listas de Prioridades, Tabelas de Dispersão, Processamento de Cadeias

- Aula 28: Listas de Prioridades
- Aula 29: Manipulação de Listas de Propriedades
- Aula 30: Tabelas de Dispersão
- Aula 31: Encadeamento Exterior
- Aula 32: Encadeamento Interior
- Aula 33: Árvores Digitais
- Aula 34: Processamento de Cadeias
- Aula 35: Árvores de Huffman
- Aula 36: Algoritmo de Huffman

Referências

- A.V.Aho, J.E.Hopcroft, J.D.Ullman, The Design and Analysis of Computer Algorithms, Addison Wesley, Reading, Ma, 1974
- T.H.Cormen, C.E.Leiserson, R.I.Rivest, Introduction to Algorithms, MIT Press, Cambridge, Mass. e McGraw-Hill, New York, NY, 1990
- C.Froidevaux, M.-C.Graudel, M.Sorla, Types de Donnés et Algorithmes, McGrawHill, 1990
- D.E.Knuth, The Art of Computer Programming 1: Fundamental Algorithms, Addison Wesley, Reading, Ma, 1968
- D.E.Knuth, The Art of Computer Programming 3: Sorting and Searching, Addison Wesley, Reading, Ma, 1973

Referências em Língua Portuguesa

- J.L.Szwarcfiter e L.Markenzon, Estrutura de Dados e seus Algoritmos, LTC Editora, Rio de Janeiro, RJ, 1994
- R.E. Torada, Desenvolvimento de Algoritmos e Estruturas de Dados, McGraw-Hill e Makron do Brasil, São Paulo, SP, 191
- P.A.Veloso, C.S.Santos, P.Azevedo, A.L.Furtado, Estruturas de Dados, Editora Campus, Rio de Janeiro, RJ, 1983
- N.Ziviani, Projeto de Algoritmos com Implementação em PASCAL, Pioneira Informática, São Paulo, SP, 1993

Livro Texto

- J.L.Szwarcfiter e L.Markenzon,

  Estrutura de Dados e seus Algoritmos,

  LTC Editora, Rio de Janeiro, RJ, 1994

Aula 1: Introdução

- Apresentação dos algoritmos
- Algoritmo de inversão de uma seqüência
- Algoritmo de cálculo de fatorial

O Conceito de Algoritmo

O Conceito de Algoritmo

- Histórico
- Idéia intuitiva: processo sistemático de resolver problemas

- Problemas básicos no estudo de algoritmos:
  - correção
  - análise

Dados

- Entrada e saída constituída de dados
- Representação de dados no computador
- Manipulação de Dados

Estruturas de Dados

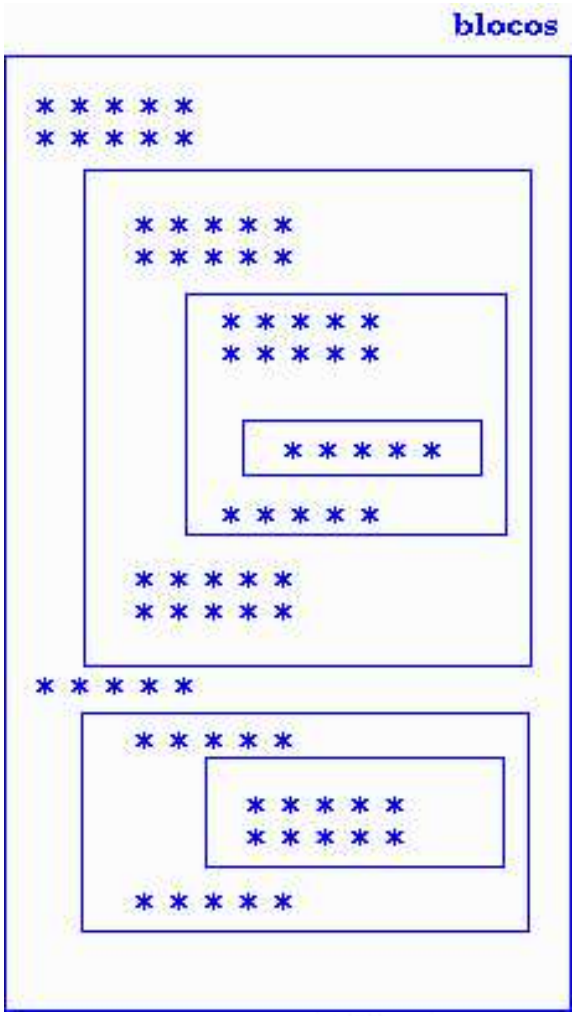
Apresentação dos Algoritmos

Apresentação dos algoritmos

- linguagens
- uso: linguagem tipo PASCAL, com livre formato

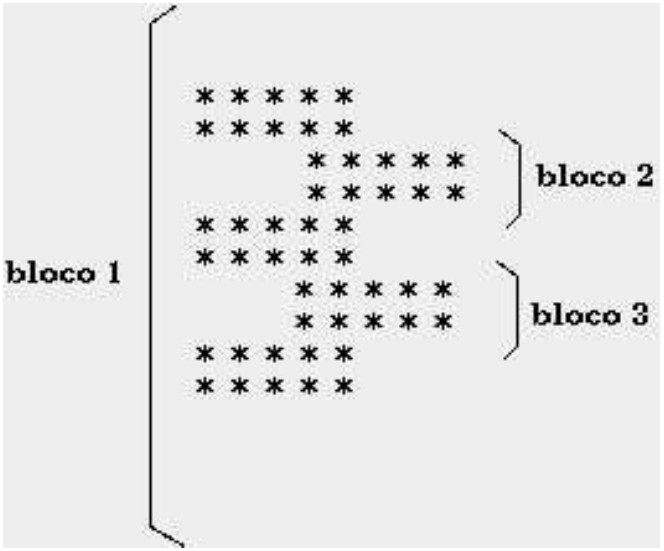
Descrição da Linguagem

- Algoritmo dividido em blocos
- Os blocos correspondem a trechos contíguos do algoritmo, tais que dois blocos são sempre:
  - disjuntos, ou
  - um deles contém o outro



Apresentação dos Algoritmos

- Na escrita, os blocos são determinados por identação, ou seja, pelo alinhamento das margens esquerdas, conforme o exemplo:



- bloco 1 contém blocos 2 e 3
- blocos 2 e 3 são disjuntos

Linguagem Utilizada

- Variáveis simples
  - A, B
    - Exemplo: `i`, `j`
- Vetores
  - `x[i]`
    - Exemplo: `x[5]`, corresponde ao 5° elemento do vetor `x`
- Matrizes
  - `x[i,j]`
    - Exemplo: `x[i,3]` é o elemento identificado pelos índices (i, 3) da matriz `x`

Linguagem Utilizada

Registros

- A localização de um registro é realizada através de um ponteiro.
- Um registro consiste de um conjunto de dados, denominados campos. Cada campo possui um nome, que o identifica.
- Um ponteiro é indicado pelo símbolo ↑
  - Exemplo: `pt↑.info` representa o campo `info` de um registro alocado no endereço contido em `pt`.

Linguagem Utilizada

- Procedimentos
  - `proc A`
    - um procedimento é chamado através de uma referência a seu nome

exemplo

```
* * * * *
A
* * * * *
```

- Funções
  - `função A(B)`
    - B corresponde ao(s) parâmetros da função
    - os parâmetros podem ser associados a "entrada" e "saída" da função
    - a função é chamada através de uma referência a seu nome

exemplo

```
* * * * *
A(B)
* * * * *
```

- Comentários
  - uma sentença iniciada por % é interpretada como comentário

Linguagem Utilizada
- Declarações -

- Declaração de atribuição
  - símbolo :=
  - A := B
    - Exemplos:

      x := 3

      x := 45.y.logz

      $x := \log^2 z$

Linguagem Utilizada
- Declarações -

- Declarações Condicionais
  - se A então B
    - Exemplos: se x - y > 54 então a := z

```
se x > y então
      x := t[y]
      i := i + 1
```

  - se A então B
    senão C
    - Exemplo: se x^2 - 3 ≠ 45 então

```
      x := y
      a := c + 1
senão i:= i +1
      x := a + i
```

Linguagem Utilizada
- Declarações -

- Declarações de Iteração
  - enquanto A faça B
    - Exemplos: enquanto $i < j^2$ faça i := i + k

      enquanto $i \neq 0$ faça
      x := a [i, j]
      i := i -1
  - para A faça B
    - Exemplos: para i = 1, 2, ..., n faça
      j := 3a - i

      para x ∈ CONJUNTO faça
      listar x

      para x ∈ C faça
      remover x de C
      x := x + 1

Linguagem Utilizada
- Declarações -

- Declarações de Iteração
  - repetir
    .
    .
    .
    até B

    - Exemplo: repetir
      listar x[i, j]
      i := i - a
      j := j + 1
      até que i < j

Linguagem Utilizada
- Declarações -

- Declaração de Parada
  - pare
    - Exemplo: se i < 0 então pare
  - Observação: paradas podem ser
    - explícitas ou
    - implícitas

Exemplo: Inversão de seqüências

- Seja uma seqüência de elementos armazenados no vetor `S[i]`, $1 \leq i \leq n$. O problema consiste em inverter a seqüência, isto é, considerá-la de trás para frente.

- `S(i)  5  2  7  1  0  6`

`S(i)  6  0  1  7  2  5`

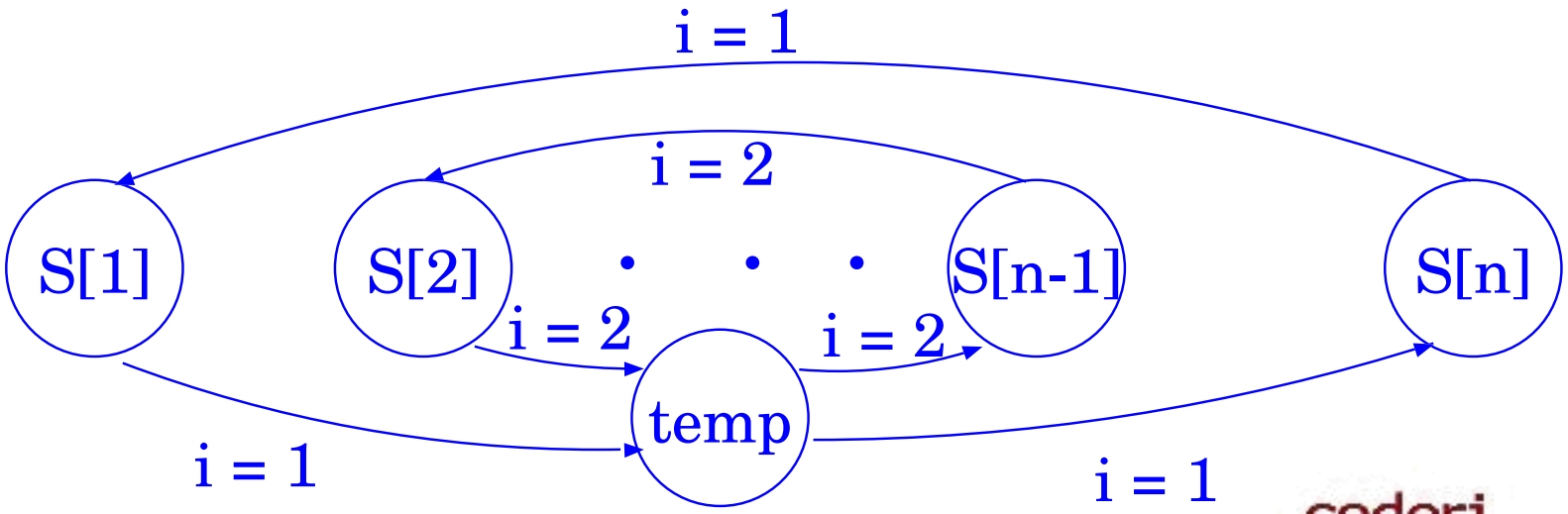
Exemplo: Inversão de seqüências

- Notações
  - Piso de x:
    $\lfloor x \rfloor$ representa o maior inteiro $\leq x$
    - Exemplos: $\lfloor 3{,}2 \rfloor = 3$
      $\lfloor 3/4 \rfloor = 0$
      $\lfloor 3 \rfloor = 3$
  - Teto de x:
    $\lceil x \rceil$ representa o menor inteiro $\geq x$
    - Exemplos: $\lceil 3{,}2 \rceil = 4$
      $\lceil 3/4 \rceil = 1$
      $\lceil 3 \rceil = 3$
  - Propriedade: $\lfloor n/2 \rfloor + \lceil n/2 \rceil = n$

Exemplo: Inversão de seqüências

Algoritmo 1.1: Inversão de uma seqüência

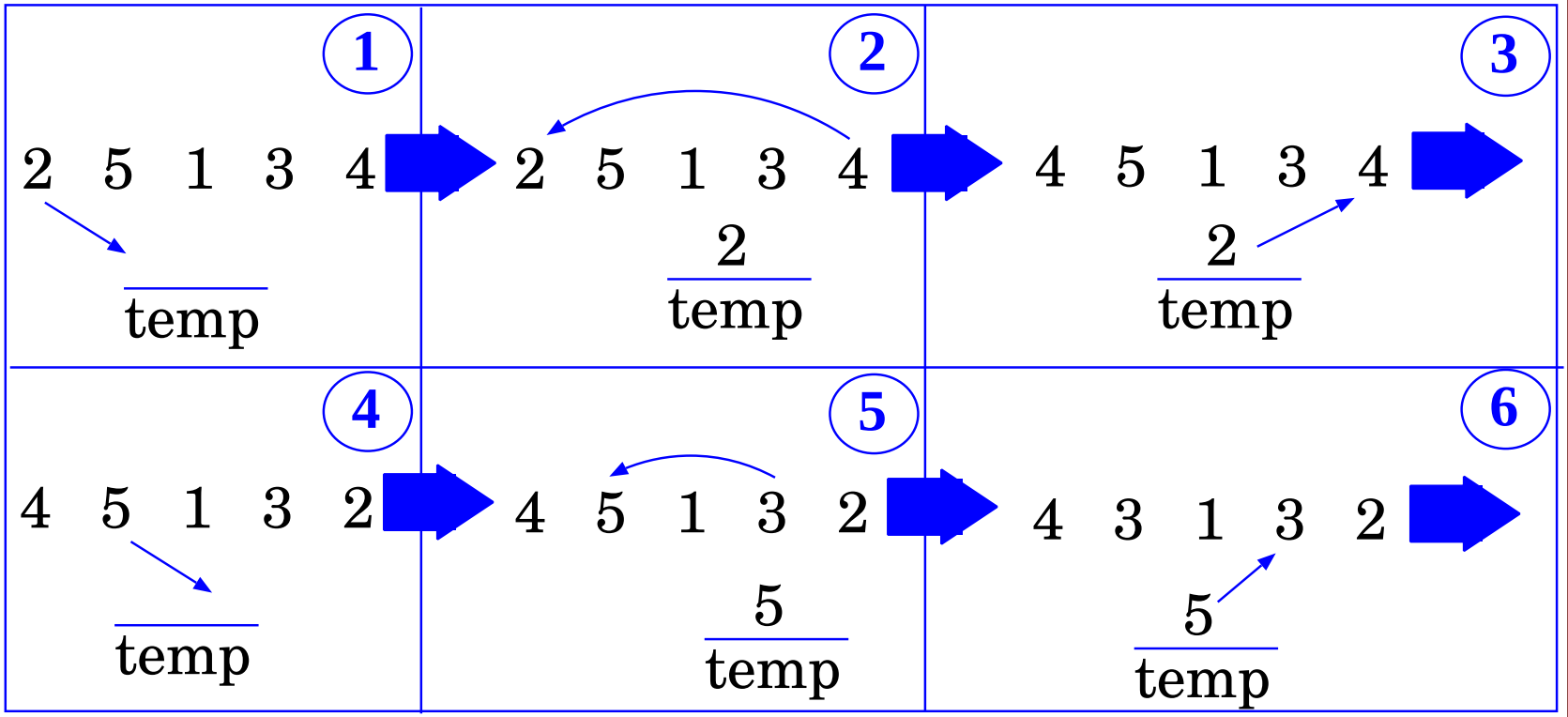
```
para i = 1, ..., ⌊n/2⌋ faça
        temp := S[ i ]
        S[ i ] := S[ n-i+1 ]
        S[ n-i+1 ] := temp
```



Exemplo: Inversão de seqüências

Exemplo:

2 5 1 3 4



4 3 1 5 2

Exemplo: Inversão de seqüências