Aula 5: Listas Lineares

- Conceito de vetores, matrizes
- Conceito de listas e deques
- Algoritmos de busca em listas lineares

Conceito

Uma lista linear agrupa informações sobre um conjunto de elementos que se relacionam entre si.

Exemplos:

- Notas de alunos de uma turma
- Quantidades de produtos no estoque de uma loja
- Nomes de clientes de uma empresa

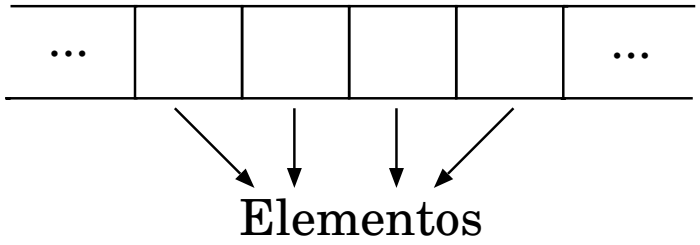
Operações básicas em listas lineares

- Busca de um elemento
- Inserção de um elemento
- Remoção de um elemento

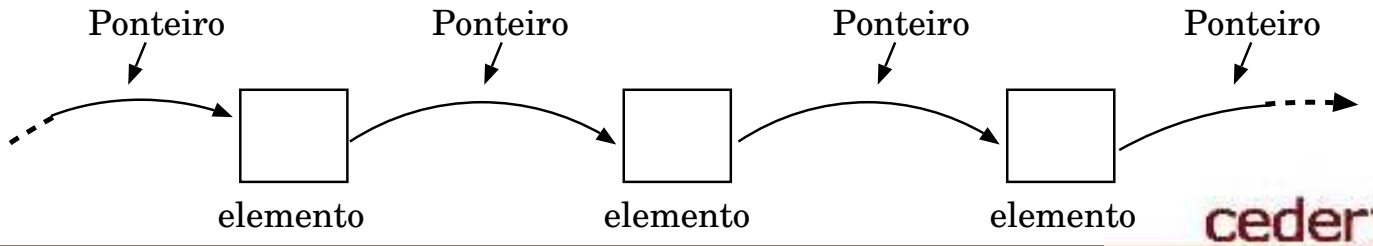
Classificação de Listas Lineares

Com relação ao modo de armazenamento da memória

- Listas Sequenciais: os elementos ocupam posições contíguas na memória.



- Listas Encadeadas: os elementos encontram-se dispersos pela memória, sendo ligados por variáveis - ponteiro.

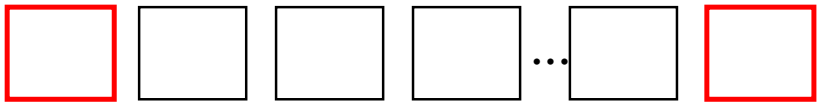


Classificação de Listas Lineares

Com relação ao modo como os elementos são inseridos/removidos

1. Listas (em geral): inserções e remoções são permitidas em qualquer posição.

2. Deques: inserções e remoções são permitidas apenas nas extremidades.

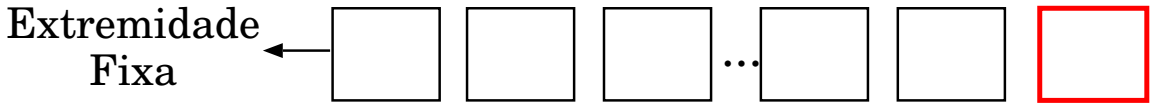


Inserir Remover Inserir Remover

Classificação de Listas Lineares

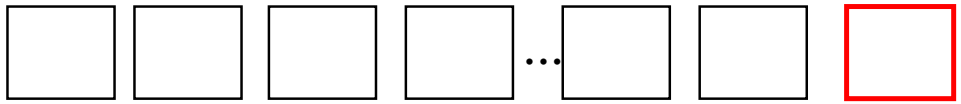
Com relação ao modo como os elementos são inseridos/removidos

- 3. Pilhas: inserções e remoções são permitidas apenas em uma das extremidades (a outra permanece fixa).



Inserir Remover

- 4. Filas: as inserções são feitas numa extremidade, e as remoções na outra.



Inserir Remover Voltar

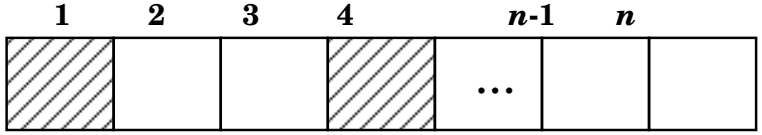
Classificação das Listas Sequenciais

Com relação ao modo de indexar os elementos

⇨ Vetores: utilizam apenas um índice para localizar o elemento desejado.

- Exemplo: Vetor V com n elementos ($n > 0$)

V

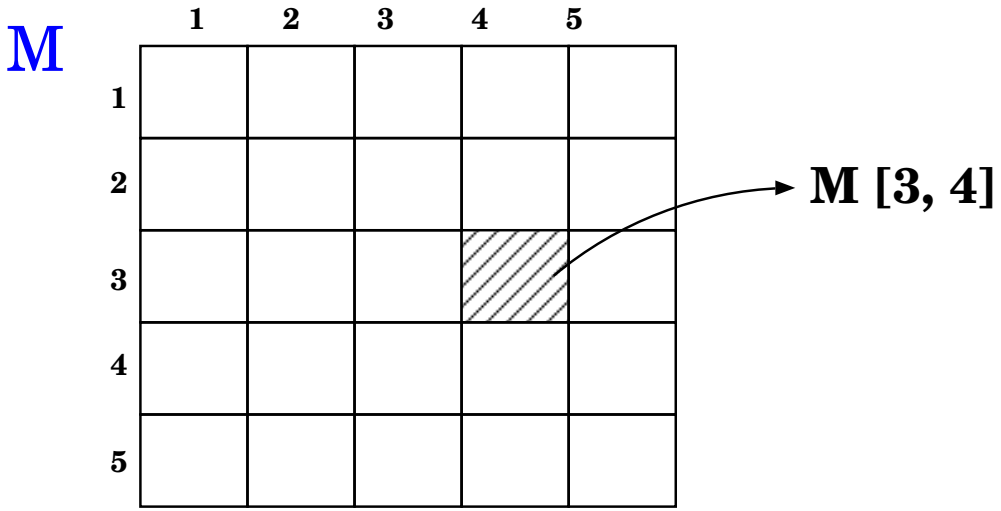


- V[1] é o primeiro elemento
- V[4] é o quarto elemento
- Em geral, V[k] é o k-ésimo elemento ($1 \leq k \leq n$)

Classificação das Listas Sequenciais

Matrizes: utilizam dois índices para localizar o elemento desejado.

- Exemplo: Matriz M com 5 linhas e 5 colunas



- M[3, 4] é o elemento que está na 3ª linha e na 4ª coluna
- Em geral, M[*i*, *j*] é o elemento que está na *i*-ésima linha e na *j*-ésima coluna

Algoritmos de Busca em Listas Sequenciais ("Busca Linear")

Método 1

- Algoritmo: Busca de um elemento no vetor V com n elementos.

```
função BUSCA1 (x)
    i := 1,  BUSCA1 := 0;
    enquanto i ≤ n faça
        (*) ..... se  v[i] = x
                  então BUSCA1 := i    % chave encontrada
                        i := n + 1
                  senão i := i + 1    % pesquisa prossegue
```

- Saída: Índice do elemento procurado, ou zero caso não seja encontrado.

Algoritmos de Busca em Listas Sequenciais ("Busca Linear")

- Complexidade do Algoritmo: número de comparações na linha marcada com (*).

  Melhor caso: 1 comparação (ocorre quando V[1] = x)

  Pior caso: n comparações

⇨ Exercício: Descrever situações em que o algoritmo acima executa o pior caso.

Tempo: 3 minutos

Solução

- Há duas situações em que o algoritmo do Método 1 executa o pior caso:
  - Situação 1: Ocorre quando x não pertence à lista
  - Situação 2: Ocorre quando V[n] = x
- Em ambos os casos, o algoritmo é forçado a fazer n comparações na linha marcada com (*).

Algoritmos de Busca em Listas Sequenciais

Método 2 (com uso de "sentinela")

- Algoritmo: Busca de um elemento no vetor V com n elementos.

```
função BUSCA2 (x)
      i := 1;  BUSCA2 := 0;
      V[n + 1] := x;                    % sentinela
(#).. enquanto  V[i] ≠ x  faça
         i := i + 1;
      se  i ≠ n + 1 então
         BUSCA2 := i
```

Algoritmos de Busca em Listas Sequenciais

- Complexidade do Algoritmo: número de comparações efetuadas pelo enquanto na linha marcada com (#).

Exercício: Descrever situações em que o algoritmo acima executa o melhor e o pior caso.

Tempo: 3 minutos

Solução

- Melhor caso: ocorre quando V[1] = x. Neste caso, o enquanto é abandonado na primeira comparação.

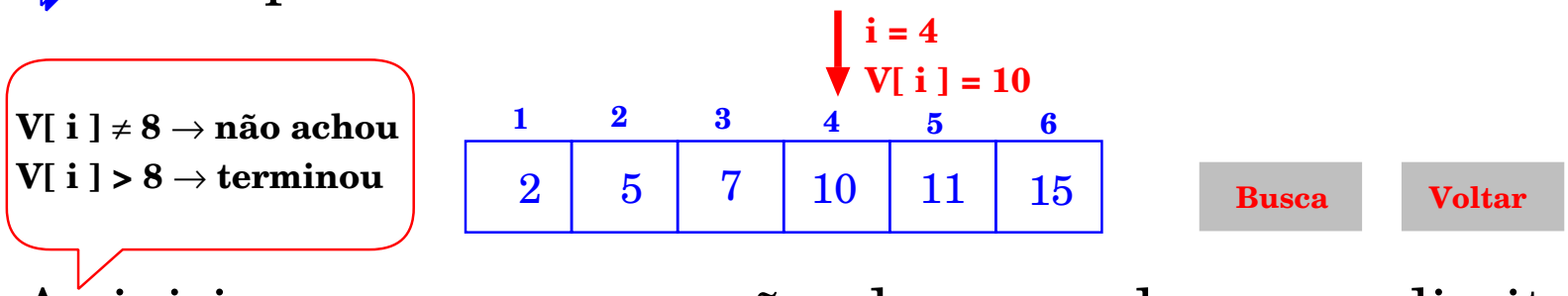
- Pior caso: ocorre quando x não pertence à lista, isto é, x não ocorre nas n primeiras posições.
  Neste caso, a sentinela garante que V[n+1] = x, e são efetuadas n+1 comparações na linha marcada com (#).

Algoritmos de Busca em Listas Sequenciais

Método 3 (Somente para listas ordenadas - "Busca Linear Ordenada")

Idéia: Quando a lista já se encontra ordenada, podemos encerrá-la quando sabemos ser inútil continuar.

Exemplo: Procurar o elemento $x = 8$ na lista abaixo:



Ao iniciarmos as comparações da esquerda para a direita, a partir do primeiro elemento, podemos encerrar a busca na quarta posição, pois como a lista é ordenada (crescentemente) sabemos que $x = 8$ não pode estar da quarta posição em diante.

Algoritmos de Busca em Listas Sequenciais

Método 3 (Somente para listas ordenadas)

Algoritmo: Busca de um elemento no vetor V, já ordenado, com n elementos.

```
função BUSCA_ORD (x)
    i := 1;  BUSCA_ORD := 0;
    V[n+1] := x;                    % sentinela
    enquanto V[i] < x faça
        i := i + 1;
    se i ≤ n  então
        BUSCA_ORD := i
```

Algoritmos de Busca em Listas Sequenciais

Exercício

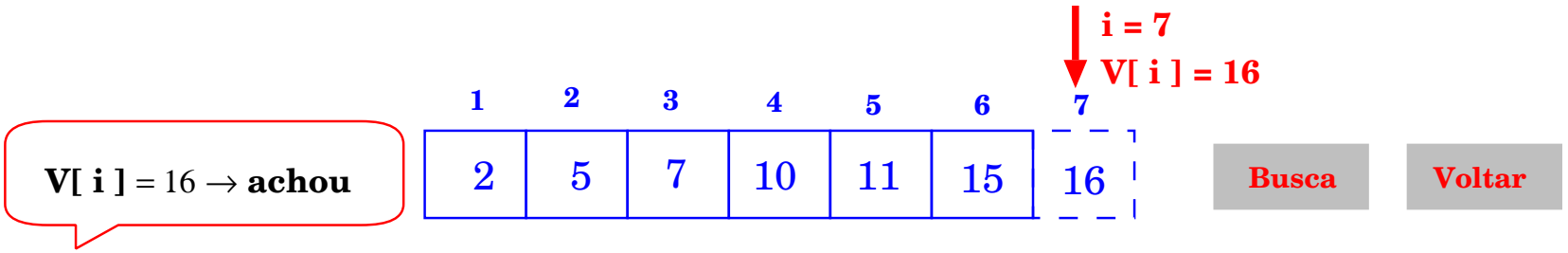
- Responda: A melhoria deste algoritmo em relação ao Método 2 significa redução do número de comparações no pior caso?

Tempo: 1 minuto

Algoritmos de Busca em Listas Sequenciais

- Apesar de o Método 3 tirar proveito do fato de a lista já estar ordenada, no pior caso continuam sendo necessárias $n + 1$ comparações.
  Este caso ocorre quando o elemento procurado é maior do que todos os presentes na lista.

- Exemplo: Procurar o elemento $x = 16$ na lista abaixo:



Serão necessárias 7 comparações

(a sétima será feita com a sentinela)

Exercício Final

Seja o seguinte algoritmo de busca em uma lista sequencial ordenada com n elementos:

```
função BUSCA_ORD2 (x)
    se x ≤ V[n]
    então
        i := 1
        enquanto V[i] < x faça
            i := i + 1

        se V[i] ≠ x
        então BUSCA_ORD2 := 0          % não encontrado
        senão BUSCA_ORD2 := i          % encontrado
    senão
        BUSCA_ORD2 := 0                % fora da tabela
```

Compare este algoritmo com o Método 3.
Em que situação o desempenho dos dois é equivalente?
Qual é a restrição que o algoritmo acima apresenta em relação ao método 3?