Estrutura de Dados

Aula 03

Prof. Luiz Antonio Schalata Pacheco, Dr. Eng.

Instituto Federal de Santa Catarina
Câmpus Garopaba
Curso Superior de Tecnologia em Sistemas para Internet

schalata@ifsc.edu.br

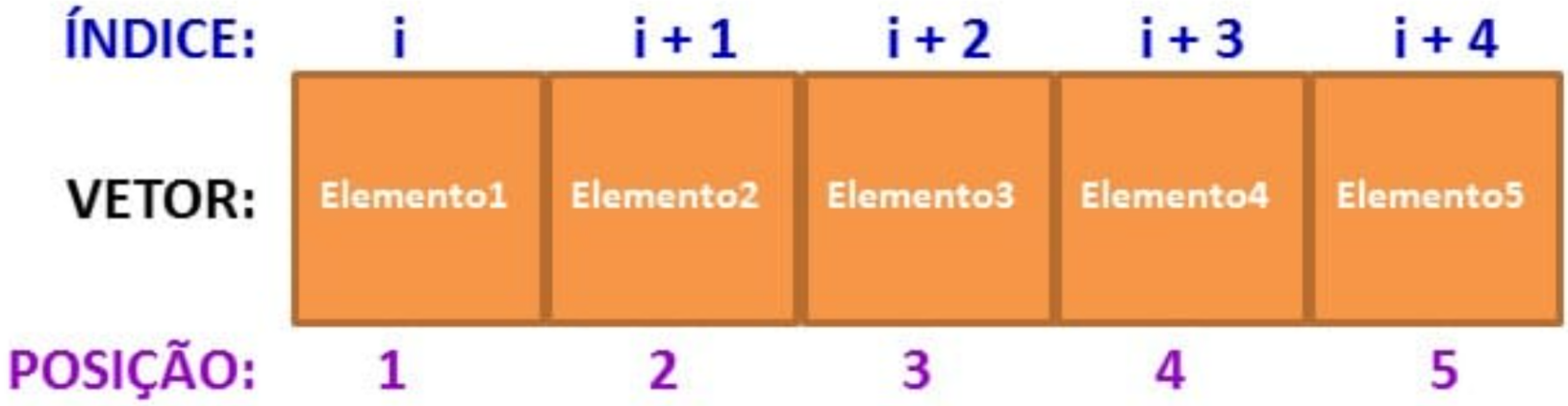
23/02/2023

Vetores Não Ordenados

Conceito de vetor

- Um vetor (*array*) é uma estrutura de dados que armazena uma sequência homogênea de itens
- São diversas variáveis agrupadas sobre o mesmo nome, facilitando o seu acesso
- O conteúdo de um vetor é acessado por meio de índices, pois é visto como posições de memória alocadas continuamente

Vetor de uma dimensão



Utilizando vetores

- Em Python é comum o uso da classe *list*

```
# Utilizacao a partir de listas
vetor = [10, 11, 12]
print(vetor)
print(vetor[2])
```

- Existe a classe *array*, com capacidade de conter apenas elementos homogêneos

```
# Usando a classe array do Python
from array import array

vetor2 = array('i', [10, 11, 12]) # Array de inteiros
print(vetor2)
print(vetor2[2])
```

Utilizando vetores

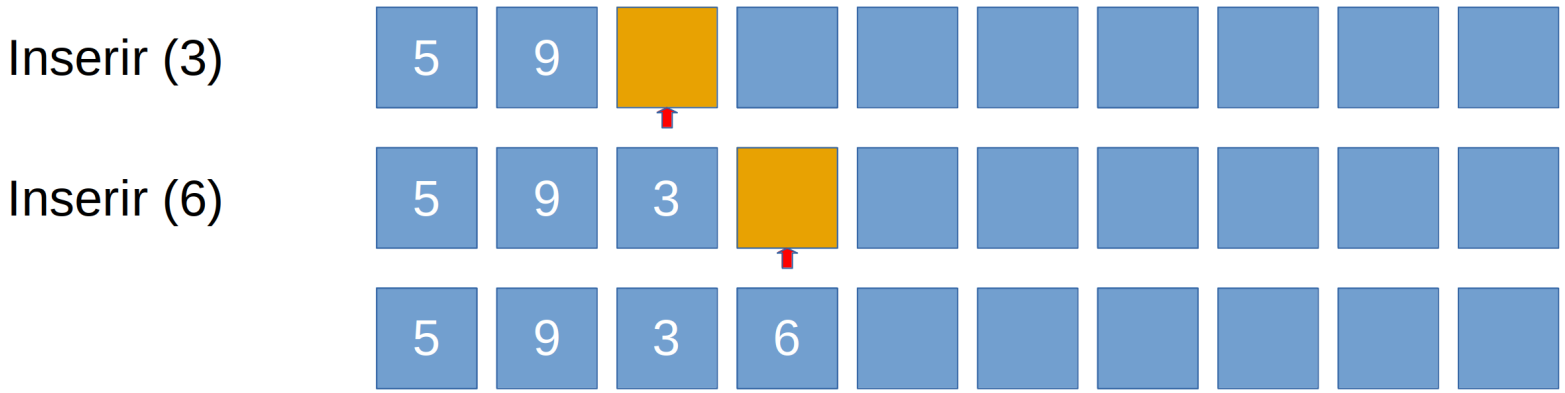
- A partir do *NumPy*

```
1 # Com o NumPy
2 import numpy as np
3
4 vetor3 = np.array([10, 11, 12])
5 print(vetor3)
6 print(vetor3[2])
```


Vetores não ordenados

- Ações que podem executadas:
  - Inserção
  - Procura
  - Remoção

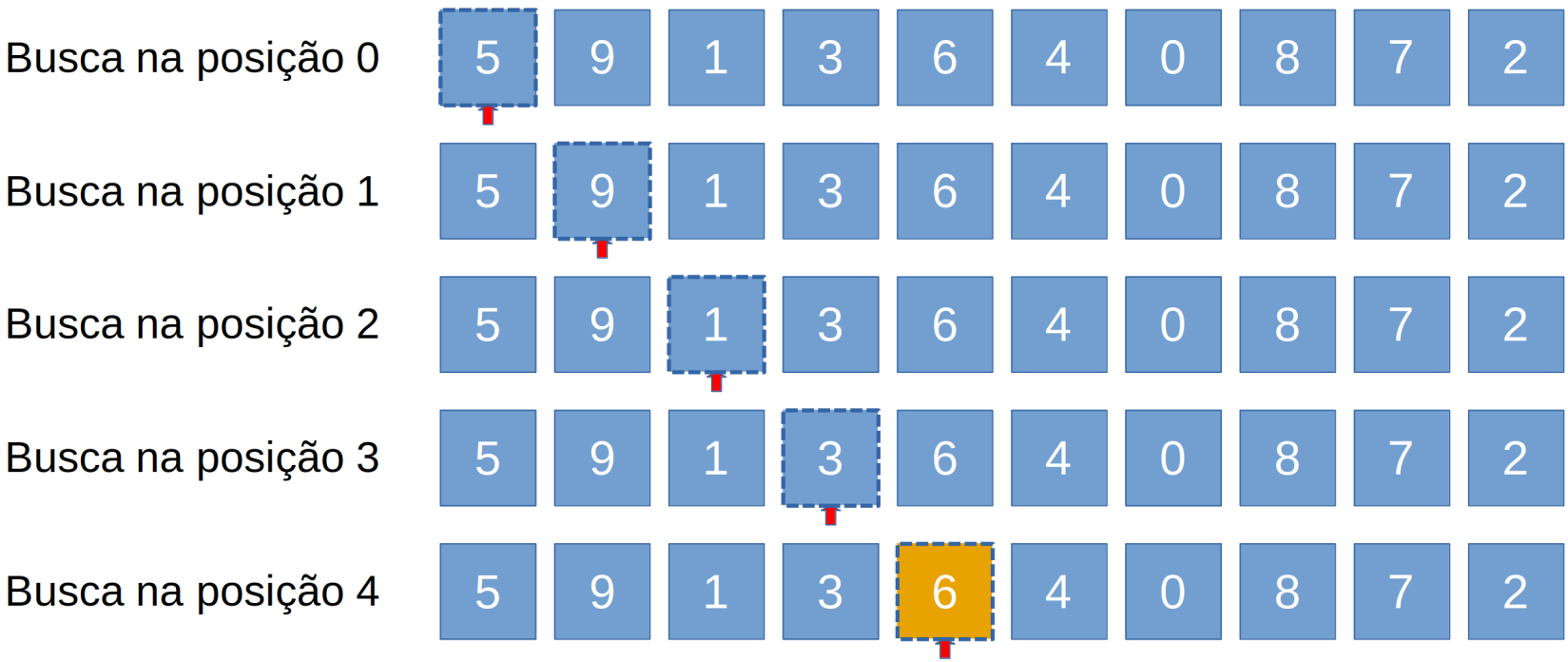
Inserção



Inserção

- Ação de um único passo, pois o dado é inserido na primeira posição vaga do vetor
- Novos elementos são sempre adicionados no final do vetor
- Na maioria das linguagens os vetores são pré-determinados com tamanho fixo
- Qual o Big(O) de uma inserção?

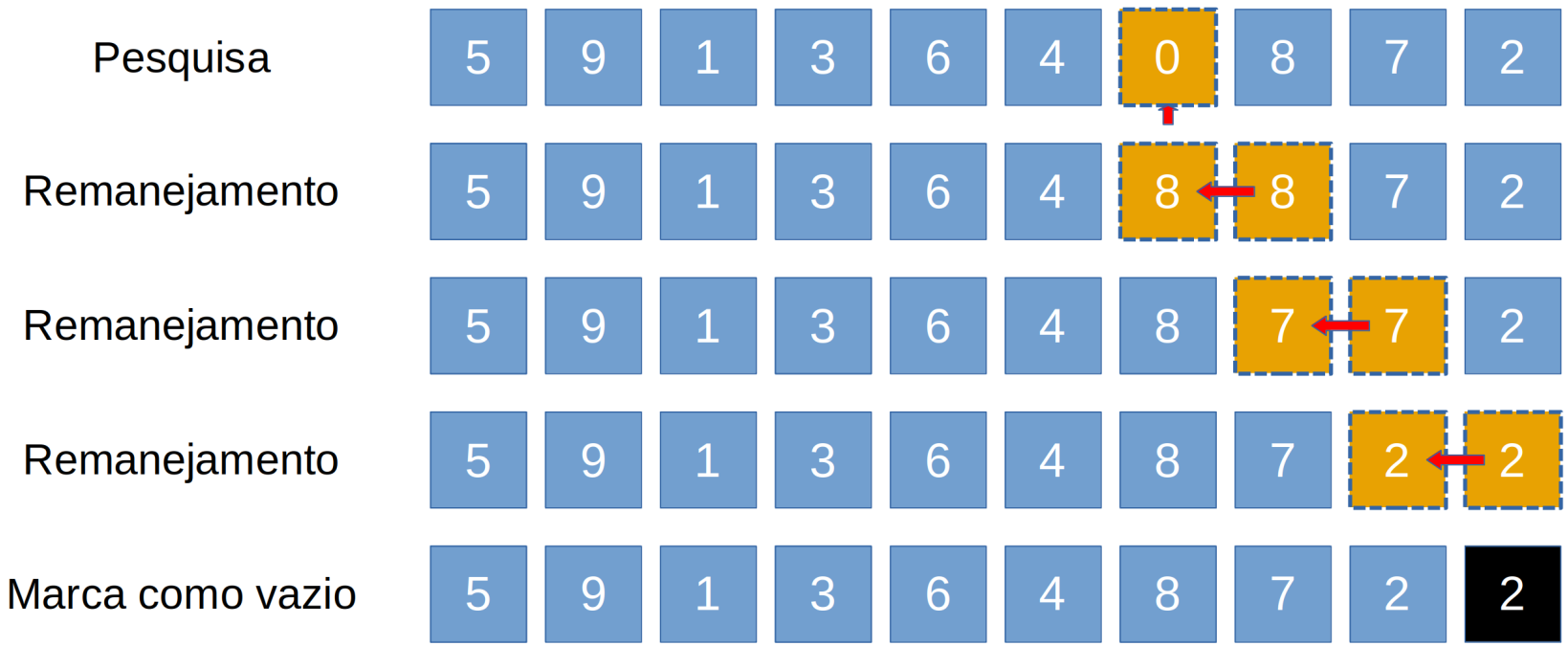
Pesquisa


Busca na posição 0
5 9 1 3 6 4 0 8 7 2
Busca na posição 1
5 9 1 3 6 4 0 8 7 2
Busca na posição 2
5 9 1 3 6 4 0 8 7 2
Busca na posição 3
5 9 1 3 6 4 0 8 7 2
Busca na posição 4
5 9 1 3 6 4 0 8 7 2

Pesquisa

- É percorrida cada posição do vetor até localizar o dado
- Melhor caso é quando o dado está na primeira posição (considerando que não há repetição dos valores!)
- Pior caso é quando está na última posição ou o dado procurado não está no vetor
- Em média, metade dos itens é pesquisado (n/2)
- Qual o Big(O) de uma pesquisa?

Exclusão


Pesquisa
5 9 1 3 6 4 0 8 7 2
Remanejamento
5 9 1 3 6 4 8 8 7 2
Remanejamento
5 9 1 3 6 4 8 7 7 2
Remanejamento
5 9 1 3 6 4 8 7 2 2
Marca como vazio
5 9 1 3 6 4 8 7 2 2

Exclusão

- O valor a ser removido é sobrescrito, remanejando os dados posteriores a ele. É feito um deslocamento a esquerda
- O último número na realidade não é apagado, mas se decrementa a variável que indica o tamanho do vetor
- Essa variável que indica o tamanho, na realidade não é exatamente uma variável setada com o tamanho, mas um ponteiro
- É uma variável que aponta para a última posição. Quando o valor é removido, ele passa a apontar para a posição anterior

Exclusão

- Primeiro é feita uma pesquisa (onde na média são pesquisados n/2 elementos), depois um remanejamento dos elementos restantes
- Mover os elementos restantes também é executado, em média, em n/2 passos
- Big(O) = O(2n) = O(n)

Considerando vetor com duplicatas

- Pesquisa: mesmo se encontrar o valor, o algoritmo terá de continuar procurando até a última posição (n passos) - O(n)
- Inserção: se não permitir dados duplicados, tem que fazer uma pesquisa primeiro. Se permitir duplicatas, basta somente inserir o dado - O(1)
- Exclusão do primeiro item: na média n/2 comparações e n/2 deslocamentos - O(n)
- Exclusão de mais itens: verificar n células e mais n/2 células - O(n)

Implementações

- Crie uma classe VetorNaoOrdenado
- O construtor deve receber um parâmetro que define a capacidade máxima do vetor
- A classe deve possuir um atributo que define a última posição ocupada do vetor e um atributo que contém os valores (tipo inteiro). Use NumPy
- Crie um método para imprimir os valores do vetor ou que informe se ele está vazio
- Crie um método para inserir um valor no vetor. Este método deve avisar se a capacidade máxima do vetor for atingida

Implementação da classe VetorNaoOrdenado

```python
import numpy as np

class VetorNaoOrdenado:
  # Construtor
  def __init__(self, capacidade):
    self.capacidade = capacidade
    self.ultima_posicao = -1
    # Cria um array vazio de inteiros
    self.valores = np.empty(self.capacidade, dtype=int)

  # Metodo para imprimir os valores - O(n)
  def imprime(self):
    if self.ultima_posicao == -1:
      print("Vetor vazio!")
    else:
      for i in range(self.ultima_posicao + 1):
        print(f'{i} - {self.valores[i]}')
```


Método de inserção

```
1  # Metodo para inserir um valor no vetor - O(1)
2  def insere(self, valor):
3    if self.ultima_posicao == self.capacidade - 1:
4      print('Capacidade maxima atingida!')
5    else:
6      self.ultima_posicao += 1
7      self.valores[self.ultima_posicao] = valor
```


Método de pesquisa

```
1  # Metodo para pesquisar valor - O(n)
2  def pesquisar(self, valor):
3      for i in range(self.ultima_posicao + 1):
4          if valor == self.valores[i]:
5              return i
6      return -1
```


Método de exclusão

```
1   # Metodo para excluir um valor - O(n)
2   def excluir(self, valor):
3     # Encontra a posicao do valor, usando o metodo
      pesquisar ja implementado
4     posicao = self.pesquisar(valor)
5     if posicao == -1:
6       return -1 # Se nao encontrou retorna -1
7     else:
8       # Faz o remanejamento dos valores
9       for i in range(posicao, self.ultima_posicao):
10        self.valores[i] = self.valores[i+1]
11      self.ultima_posicao -= 1
```


Exemplo do programa principal

```
1  # Exemplo do main
2  from VetorNaoOrdenado import VetorNaoOrdenado
3
4  if __name__ == '__main__':
5    vetor = VetorNaoOrdenado(5)
6    vetor.imprime()
7    vetor.insere(2)
8    vetor.insere(3)
9    vetor.insere(8)
10   vetor.insere(7)
11   vetor.insere(11)
12   vetor.imprime()
13   vetor.insere(4)
14   print(vetor.pesquisar(3))
15   vetor.excluir(2)
16   vetor.imprime()
```

Exercícios

- Considere o vetor não ordenado:

- Considerando não haver duplicatas, quantos passos são necessários para incluir o número 11?
- E com a possibilidade de haver duplicatas?
- Quantos passos para pesquisar o número 89? E se permitir duplicatas?
- Quantos passos para pesquisar o número 100?
- Quantos passos para remover o número 9?