

# CI1055: Algoritmos e Estruturas de Dados I

Profs. Drs. Marcos Castilho, Bruno Müller Jr, Carmem Hara

Departamento de Informática/UFPR

20 de agosto de 2020

Resumo

Vetores como estruturas de dados

# Objetivos da aula

- Caracterizar o que é uma estrutura de dados
- Entender como usar vetores como estruturas de dados

# Vetores como estruturas de dados

- Os vetores, por si só, são um tipo estruturado da linguagem *Pascal*
- Uma estrutura de dados possui algo adicional:
  - uma propriedade que é relativa ao uso que o programador faz do espaço de memória disponível
- Esta propriedade pode, e deve, ser explorada na elaboração do algoritmo

- Um vetor de no máximo 200 elementos
- Propriedade:
  - os elementos do vetor não têm ordem relativa
  - pode haver repetição de elementos
  - eles são inseridos do início ao fim

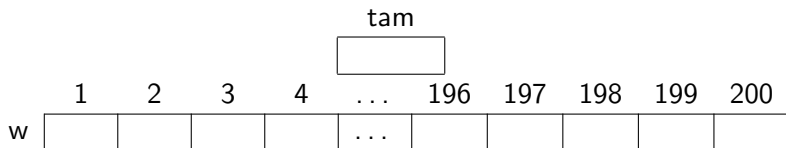
- Um vetor de no máximo 200 elementos
- Propriedade:
  - os elementos do vetor não têm ordem relativa
  - pode haver repetição de elementos
  - eles são inseridos do fim ao início

- Um vetor de no máximo 200 elementos
- Propriedade:
  - os elementos do vetor devem estar em ordem crescente
  - não pode haver repetição de elementos
  - eles são inseridos do início ao fim

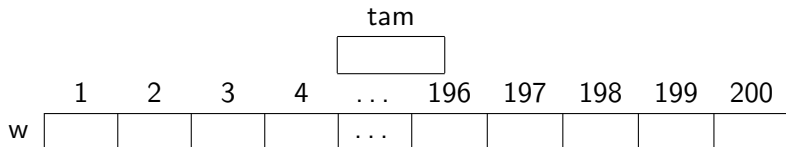
- Inserir e remover elementos é diferente nas estruturas de dados 1, 2 e 3 acima
- Estas operações devem levar em consideração a propriedade definida para a estrutura
- Mas isso é tópico para outras aulas
- Veremos somente um exemplo básico, que é inserir e remover nas estruturas de dados 1 e 2, acima definidas

# Exemplos

Inserir os números 5, -4, 8 e 7 na estrutura de dados 1:



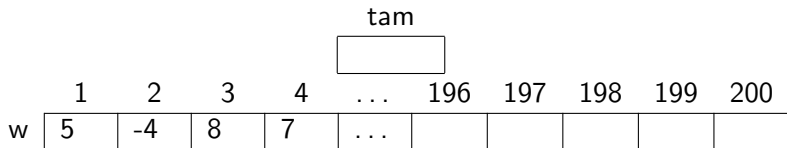
Inserir os números 5, -4, 8 e 7 na estrutura de dados 2:



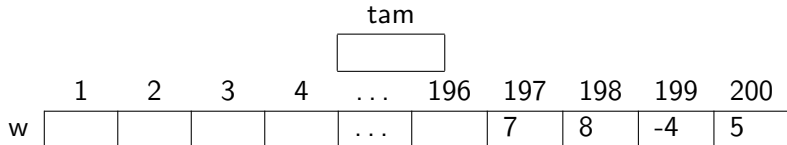


# Exemplos

Remover o número 7 da estrutura de dados 1:

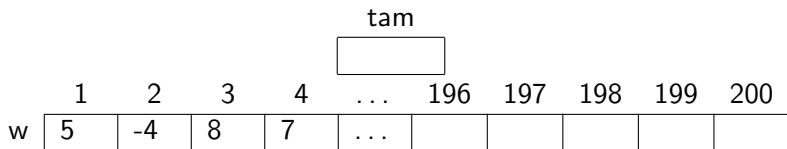


Remover o número 7 da estrutura de dados 2:

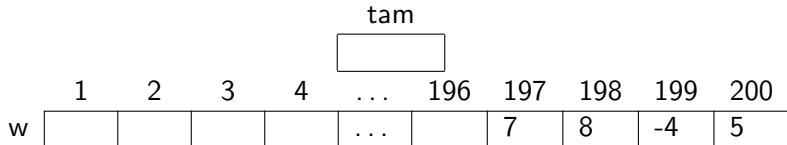


# Exemplos

Remover o número 5 da estrutura de dados 1:



Remover o número 5 da estrutura de dados 2:



# Inserir elementos

```
1  const MIN=1; MAX=200;
2  type vetor = array [MIN..MAX] of longint;
3  var v: vetor;
```

## Estrutura 1

```
1  begin
2    tam:= MIN - 1;
3    read (n);
4    while n <> 0 do
5      begin
6        tam:= tam + 1;
7        v[tam]:= n;
8        read (n);
9      end;
10 end.
```

## Estrutura 2

```
1  begin
2    tam:= MAX + 1;
3    read (n);
4    while n <> 0 do
5      begin
6        tam:= tam - 1;
7        v[tam]:= n;
8        read (n);
9      end;
10 end.
```

# Remover o último elemento

```
1  const MIN=1; MAX=200;  
2  type vetor = array [MIN..MAX] of longint;  
3  var v: vetor;
```

Estrutura 1

```
1  begin  
2      tam:= tam - 1;  
3  end.
```

Estrutura 2

```
1  begin  
2      tam:= tam + 1;  
3  end.
```

# Remover o primeiro elemento

```
1  const MIN=1; MAX=200;  
2  type vetor = array [MIN..MAX] of longint;  
3  var v: vetor;
```

## Estrutura 1

```
1  begin  
2    v[MIN]:= v[tam];  
3    tam:= tam - 1;  
4  end.
```

## Estrutura 2

```
1  begin  
2    v[MAX]:= v[tam];  
3    tam:= tam + 1;  
4  end.
```

- este material está no livro no capítulo 9, seções 9.1 até 9.3

- Slides feitos em  $\text{\LaTeX}$  usando beamer
- Licença

*Creative Commons* Atribuição-Uso Não-Comercial-Vedada a Criação de Obras Derivadas 2.5 Brasil License.<http://creativecommons.org/licenses/by-nc-nd/2.5/br/>

Creative Commons Atribuição-Uso Não-Comercial-Vedada a Criação de Obras Derivadas 2.5 Brasil License.<http://creativecommons.org/licenses/by-nc-nd/2.5/br/>