Estruturas de Dados

A. G. Silva, A. von Wangenheim, J. E. Martina

07 de agosto de 2018

Definição de Estruturas de Dados

Definição

Estruturas de Dados é a disciplina que estuda as técnicas computacionais para a organização e manipulação eficiente de quaisquer quantidades de informação.

Motivação de Estruturas de Dados

- Ao estudar estruturas de dados teremos sempre este par:
  - Um conjunto estruturado de informações usualmente uma classe de objetos ou um tipo de dados;
  - Um conjunto definido de operações sobre estes dados usualmente um conjunto de métodos ou funções.
- Em um projeto de software sempre vamos nos preocupar de qual forma estão organizados os dados: qual a sua estrutura, que tipo de operações podem ser realizadas, quais procedimentos que atuam sobre estes dados.

Conceito de Pilhas

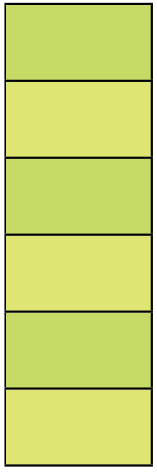
Pilha

É uma estrutura de dados cujo funcionamento é inspirado no conceito “natural” de empilhar, onde o primeiro a entrar é o último a sair (LIFO).

Conceito de Pilhas

Pilha

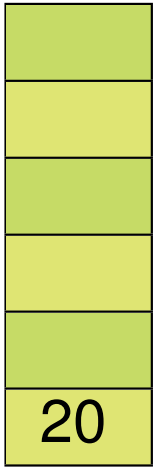
É uma estrutura de dados cujo funcionamento é inspirado no conceito “natural” de empilhar, onde o primeiro a entrar é o último a sair (LIFO).



Conceito de Pilhas

Pilha

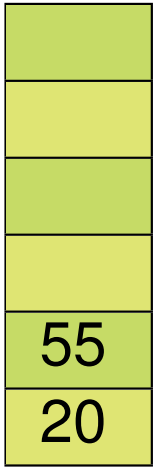
É uma estrutura de dados cujo funcionamento é inspirado no conceito “natural” de empilhar, onde o primeiro a entrar é o último a sair (LIFO).



Conceito de Pilhas

Pilha

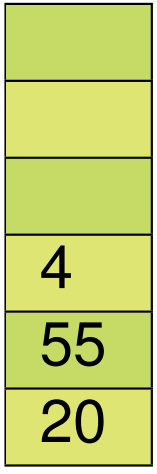
É uma estrutura de dados cujo funcionamento é inspirado no conceito “natural” de empilhar, onde o primeiro a entrar é o último a sair (LIFO).



Conceito de Pilhas

Pilha

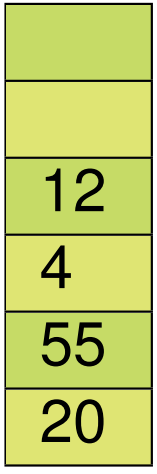
É uma estrutura de dados cujo funcionamento é inspirado no conceito “natural” de empilhar, onde o primeiro a entrar é o último a sair (LIFO).



Conceito de Pilhas

Pilha

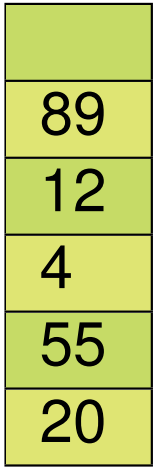
É uma estrutura de dados cujo funcionamento é inspirado no conceito “natural” de empilhar, onde o primeiro a entrar é o último a sair (LIFO).



Conceito de Pilhas

Pilha

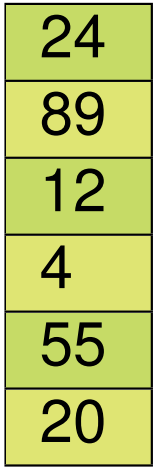
É uma estrutura de dados cujo funcionamento é inspirado no conceito “natural” de empilhar, onde o primeiro a entrar é o último a sair (LIFO).



Conceito de Pilhas

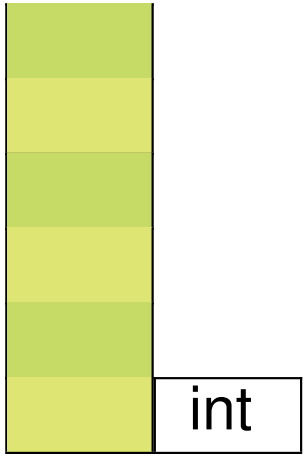
Pilha

É uma estrutura de dados cujo funcionamento é inspirado no conceito “natural” de empilhar, onde o primeiro a entrar é o último a sair (LIFO).



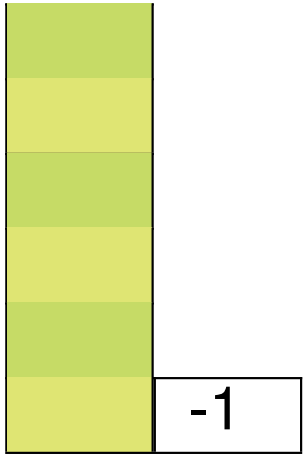
Pilhas em vetor

- Vetores possuem um espaço limitado para armazenar dados;
- Necessitamos definir um espaço grande o suficiente para a nossa pilha;
- Necessitamos de um indicador de qual elemento do vetor é o atual topo da pilha.



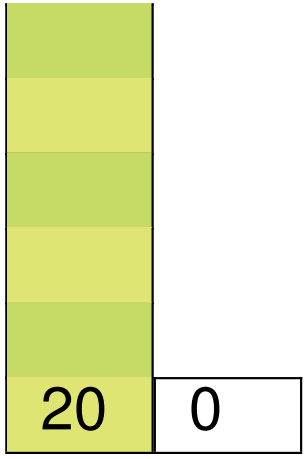
Pilhas em vetor

- Vetores possuem um espaço limitado para armazenar dados;
- Necessitamos definir um espaço grande o suficiente para a nossa pilha;
- Necessitamos de um indicador de qual elemento do vetor é o atual topo da pilha;
- Pilha vazia!



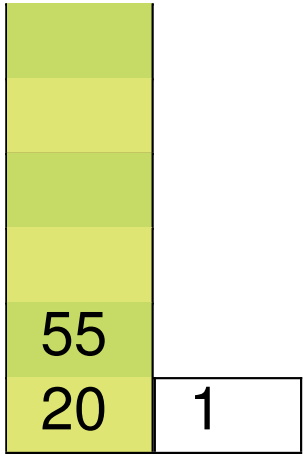
Pilhas em vetor

- Vetores possuem um espaço limitado para armazenar dados;
- Necessitamos definir um espaço grande o suficiente para a nossa pilha;
- Necessitamos de um indicador de qual elemento do vetor é o atual topo da pilha.



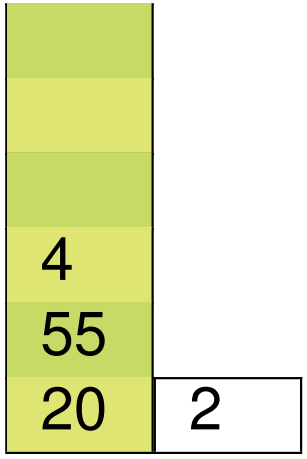
Pilhas em vetor

- Vetores possuem um espaço limitado para armazenar dados;
- Necessitamos definir um espaço grande o suficiente para a nossa pilha;
- Necessitamos de um indicador de qual elemento do vetor é o atual topo da pilha.



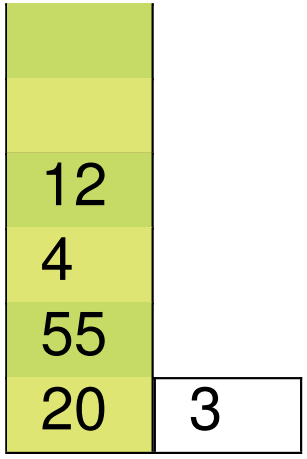
Pilhas em vetor

- Vetores possuem um espaço limitado para armazenar dados;
- Necessitamos definir um espaço grande o suficiente para a nossa pilha;
- Necessitamos de um indicador de qual elemento do vetor é o atual topo da pilha.



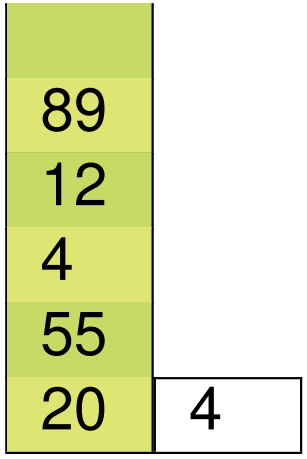
Pilhas em vetor

- Vetores possuem um espaço limitado para armazenar dados;
- Necessitamos definir um espaço grande o suficiente para a nossa pilha;
- Necessitamos de um indicador de qual elemento do vetor é o atual topo da pilha.



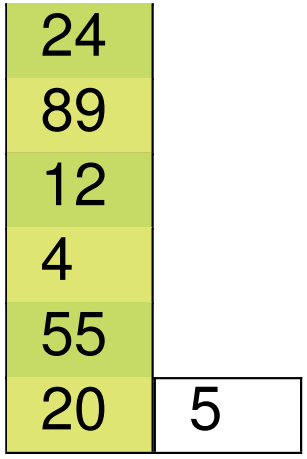
Pilhas em vetor

- Vetores possuem um espaço limitado para armazenar dados;
- Necessitamos definir um espaço grande o suficiente para a nossa pilha;
- Necessitamos de um indicador de qual elemento do vetor é o atual topo da pilha.



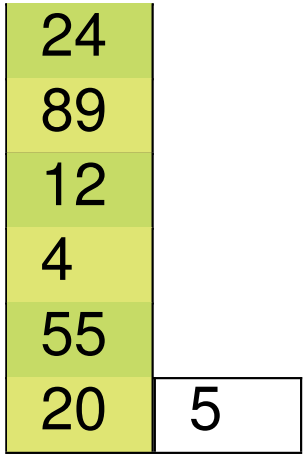
Pilhas em vetor

- Vetores possuem um espaço limitado para armazenar dados;
- Necessitamos definir um espaço grande o suficiente para a nossa pilha;
- Necessitamos de um indicador de qual elemento do vetor é o atual topo da pilha.



Pilhas em vetor

- Vetores possuem um espaço limitado para armazenar dados;
- Necessitamos definir um espaço grande o suficiente para a nossa pilha;
- Necessitamos de um indicador de qual elemento do vetor é o atual topo da pilha;
- Pilha cheia!



Modelagem da Pilha

- Aspecto estrutural:
  - Necessitamos de um vetor para armazenar as informações;
  - Necessitamos de um indicador da posição atual do topo da pilha.

```
constante MAXPILHA ← 100;

classe Pilha {
    T _dados[MAXPILHA];  // Vetor estatico
    inteiro _topo;
};
```

Modelagem da Pilha

- Aspecto estrutural:
  - Necessitamos de um vetor para armazenar as informações;
  - Necessitamos de um indicador da posição atual do topo da pilha.

```
constante MAXPILHA ← 100;

classe Pilha {
    T* _dados;  // _dados ← new T[MAXPILHA];
    inteiro _topo;
    inteiro _tam ← MAXPILHA;
};
```

Modelagem da Pilha

- Aspecto funcional:
  - Temos que inserir e remover dados da pilha;
  - Temos que testar se a pilha está vazia ou cheia;
  - Temos que inicializar a pilha.

Modelagem da Pilha

- Inicializar ou limpar:
  - `Pilha();`
  - `Pilha(int tam);`
  - `limpaPilha();`
- Testar se a pilha está vazia ou cheia:
  - `bool pilhaCheia();`
  - `bool pilhaVazia();`
- Inserir e remover dados da pilha:
  - `empilha(T dado);`
  - `T desempilha();`
  - `T topo();`

Método `Pilha()`

```
Pilha()
inicio
      _dados ← new T[_tam];
      _topo ← -1;
fim
```

Método `Pilha(int tam)`

```
Pilha(int tam)
inicio
      _tam ← tam;
      _dados ← new T[tam];
      _topo ← -1;
fim
```

Método `limpaPilha()`

```
void limpaPilha()
inicio
      _topo ← -1;
fim
```

Método `pilhaCheia()`

```
bool pilhaCheia()
inicio
    SE (_topo = _tam - 1) ENTAO
        RETORNE(Verdadeiro);
    SENAO
        RETORNE(Falso);
fim
```

Método `pilhaVazia()`

```
bool pilhaVazia()
inicio
    SE (_topo = -1) ENTAO
        RETORNE(Verdadeiro)
    SENAO
        RETORNE(Falso);
fim
```

Método empilha(T dado)

```
void empilha(T dado)
inicio
    SE (pilhaCheia) ENTAO
        JogueExcecao(ERROPILHACHEIA);
    SENAO
        _topo ← _topo + 1
        _dados[_topo] ← dado;
    FIM SE
fim
```

Método `T desempilha()`

```
T desempilha()
inicio
    SE (pilhaVazia) ENTAO
        JogueExcecao(ERROPILHAVAZIA);
    SENAO
        _topo ← _topo - 1;
        RETORNE(_dados[_topo+1]);
    FIM SE
fim
```

Método `T topo()`

```
T topo()
inicio
    SE (pilhaVazia) ENTAO
        JogueExcecao(ERROPILHAVAZIA);
    SENAO
        RETORNE(_dados[_topo]);
    FIM SE
fim
```

Trabalho de Pilha em vetor

- Implemente uma classe Pilha todas as operações vistas;
- Implemente a pilha usando *templates*;
- Implemente a pilha com um número de elementos variável definido na instanciação;
- Use as melhores práticas de orientação a objetos;
- Documente todas as classes, métodos e atributos;
- Aplique os testes unitários disponíveis no Moodle da disciplina para validar sua estrutura de dados;
- Entregue até a data definida no Moodle.

Conceito de Filas

Fila

É uma estrutura de dados cujo funcionamento é inspirado no conceito “natural” de enfileiramento, onde o primeiro a entrar é o primeiro a sair (FIFO).

Conceito de Filas

Fila

É uma estrutura de dados cujo funcionamento é inspirado no conceito “natural” de enfileiramento, onde o primeiro a entrar é o primeiro a sair (FIFO).

Conceito de Filas

Fila

É uma estrutura de dados cujo funcionamento é inspirado no conceito “natural” de enfileiramento, onde o primeiro a entrar é o primeiro a sair (FIFO).

Conceito de Filas

Fila

É uma estrutura de dados cujo funcionamento é inspirado no conceito “natural” de enfileiramento, onde o primeiro a entrar é o primeiro a sair (FIFO).

Conceito de Filas

Fila

É uma estrutura de dados cujo funcionamento é inspirado no conceito “natural” de enfileiramento, onde o primeiro a entrar é o primeiro a sair (FIFO).

Conceito de Filas

Fila

É uma estrutura de dados cujo funcionamento é inspirado no conceito “natural” de enfileiramento, onde o primeiro a entrar é o primeiro a sair (FIFO).

Conceito de Filas

Fila

É uma estrutura de dados cujo funcionamento é inspirado no conceito “natural” de enfileiramento, onde o primeiro a entrar é o primeiro a sair (FIFO).

Conceito de Filas

Fila

É uma estrutura de dados cujo funcionamento é inspirado no conceito “natural” de enfileiramento, onde o primeiro a entrar é o primeiro a sair (FIFO).

Conceito de Filas

Fila

É uma estrutura de dados cujo funcionamento é inspirado no conceito “natural” de enfileiramento, onde o primeiro a entrar é o primeiro a sair (FIFO).

Conceito de Filas

Fila

É uma estrutura de dados cujo funcionamento é inspirado no conceito “natural” de enfileiramento, onde o primeiro a entrar é o primeiro a sair (FIFO).

Conceito de Filas

Fila

É uma estrutura de dados cujo funcionamento é inspirado no conceito “natural” de enfileiramento, onde o primeiro a entrar é o primeiro a sair (FIFO).

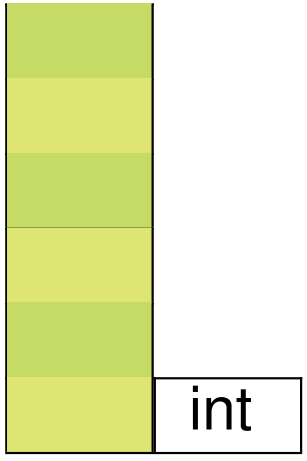
- Duas operações:
  - Inserir no fim
  - Remover do início

Conceito de Filas

- É importante para gerência de dados/processos por ordem cronológica:
  - Fila de impressão em uma impressora de rede;
  - Fila de pedidos de uma expedição ou tele-entrega;
- É importante para simulação de processos sequenciais:
  - chão de fábrica: fila de camisetas a serem estampadas;
  - comércio: simulação de fluxo de um caixa de supermercado;
  - tráfego: simulação de um cruzamento com um semáforo.

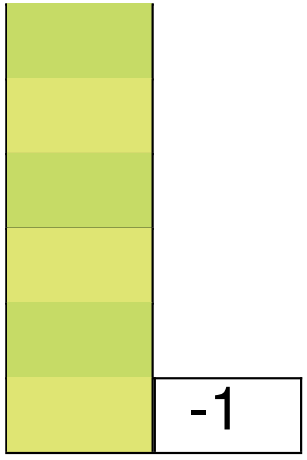
Filas em vetor

- Vetores possuem um espaço limitado para armazenar dados;
- Necessitamos definir um espaço grande o suficiente para a nossa pilha;
- Necessitamos de um indicador de qual elemento do vetor é o atual topo da pilha;
- Incluímos sempre no fim.



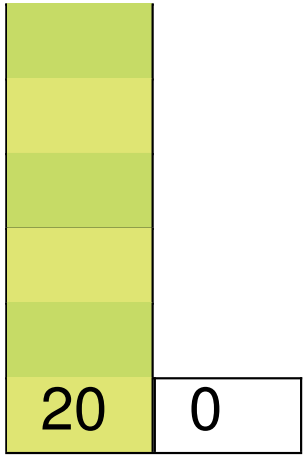
Filas em vetor

- Vetores possuem um espaço limitado para armazenar dados;
- Necessitamos definir um espaço grande o suficiente para a nossa pilha;
- Necessitamos de um indicador de qual elemento do vetor é o atual topo da pilha;
- Incluímos sempre no fim;
- Fila vazia!



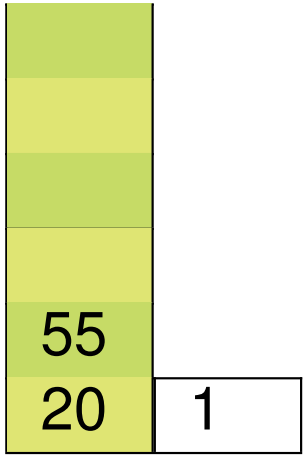
Filas em vetor

- Vetores possuem um espaço limitado para armazenar dados;
- Necessitamos definir um espaço grande o suficiente para a nossa pilha;
- Necessitamos de um indicador de qual elemento do vetor é o atual topo da pilha;
- Incluímos sempre no fim.



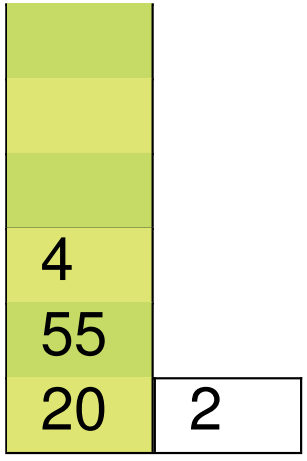
Filas em vetor

- Vetores possuem um espaço limitado para armazenar dados;
- Necessitamos definir um espaço grande o suficiente para a nossa pilha;
- Necessitamos de um indicador de qual elemento do vetor é o atual topo da pilha;
- Incluímos sempre no fim.



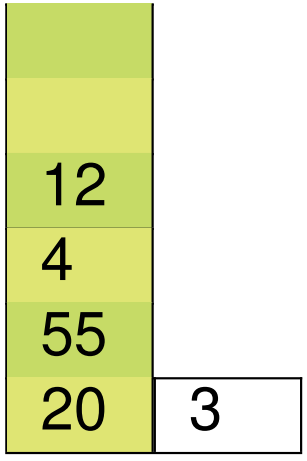
Filas em vetor

- Vetores possuem um espaço limitado para armazenar dados;
- Necessitamos definir um espaço grande o suficiente para a nossa pilha;
- Necessitamos de um indicador de qual elemento do vetor é o atual topo da pilha;
- Incluímos sempre no fim.



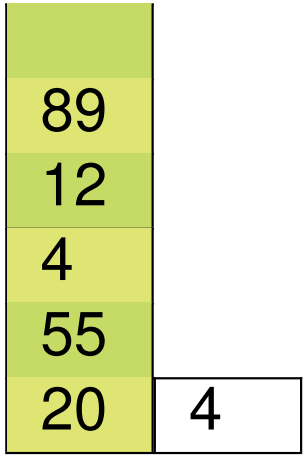
Filas em vetor

- Vetores possuem um espaço limitado para armazenar dados;
- Necessitamos definir um espaço grande o suficiente para a nossa pilha;
- Necessitamos de um indicador de qual elemento do vetor é o atual topo da pilha;
- Incluímos sempre no fim.



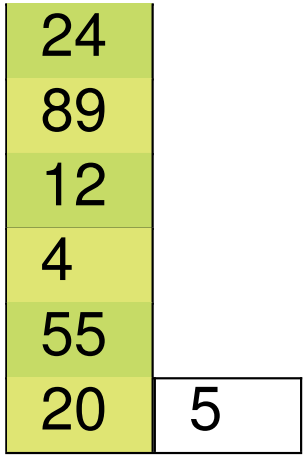
Filas em vetor

- Vetores possuem um espaço limitado para armazenar dados;
- Necessitamos definir um espaço grande o suficiente para a nossa pilha;
- Necessitamos de um indicador de qual elemento do vetor é o atual topo da pilha;
- Incluímos sempre no fim.



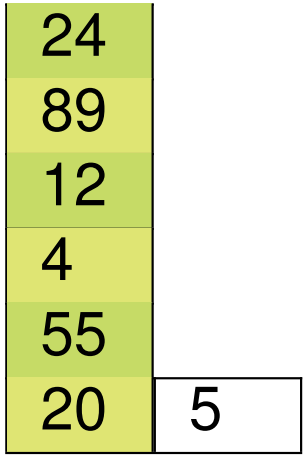
Filas em vetor

- Vetores possuem um espaço limitado para armazenar dados;
- Necessitamos definir um espaço grande o suficiente para a nossa pilha;
- Necessitamos de um indicador de qual elemento do vetor é o atual topo da pilha;
- Incluímos sempre no fim.



Filas em vetor

- Vetores possuem um espaço limitado para armazenar dados;
- Necessitamos definir um espaço grande o suficiente para a nossa pilha;
- Necessitamos de um indicador de qual elemento do vetor é o atual topo da pilha;
- Incluímos sempre no fim;
- Fila cheia!



Modelagem da Fila

- Aspecto estrutural:
  - Necessitamos de um vetor para armazenar as informações;
  - Necessitamos de um indicador da posição atual do fim da fila.

```
constante MAXFILA ← 100;

classe Fila {
    T* _dados;  // _dados ← new T[MAXFILA];
    inteiro _fim;
    inteiro _tam ← MAXFILA;
};
```

Modelagem da Fila

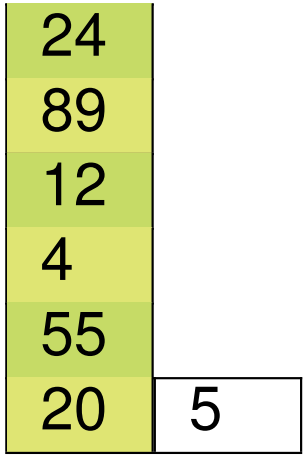
- Aspecto funcional:
  - Temos que inserir e remover dados da fila;
  - Temos que testar se a fila está vazia ou cheia;
  - Temos que inicializar a fila.

Modelagem da Fila

- Inicializar ou limpar:
  - `Fila();`
  - `Fila(int tam);`
  - `limpaFila();`
- Testar se a fila está vazia ou cheia:
  - `bool filaCheia();`
  - `bool filaVazia();`
- Inserir e remover dados da pilha:
  - `insere(T dado);`
  - `T remove();`
  - `T ultimo();`

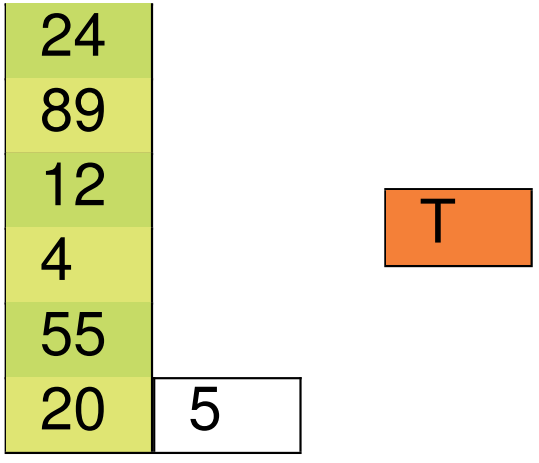
Método `T remove()`

- Testamos se há elementos;
- Decrementamos o fim da fila (último);
- Salvamos o primeiro elemento em variável auxiliar;
- Empurramos tudo para a frente.



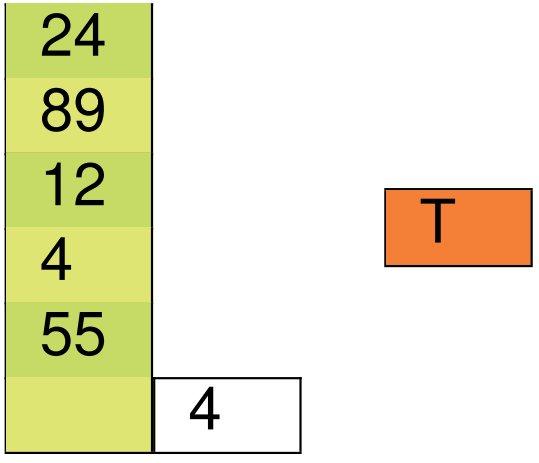
Método `T remove()`

- Testamos se há elementos;
- Decrementamos o fim da fila (último);
- Salvamos o primeiro elemento em variável auxiliar;
- Empurramos tudo para a frente.



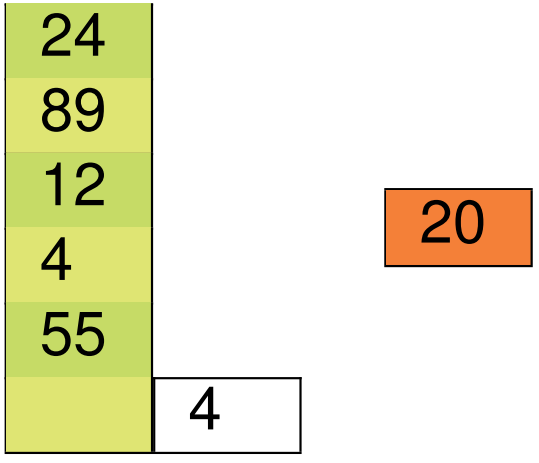
Método `T remove()`

- Testamos se há elementos;
- Decrementamos o fim da fila (último);
- Salvamos o primeiro elemento em variável auxiliar;
- Empurramos tudo para a frente.



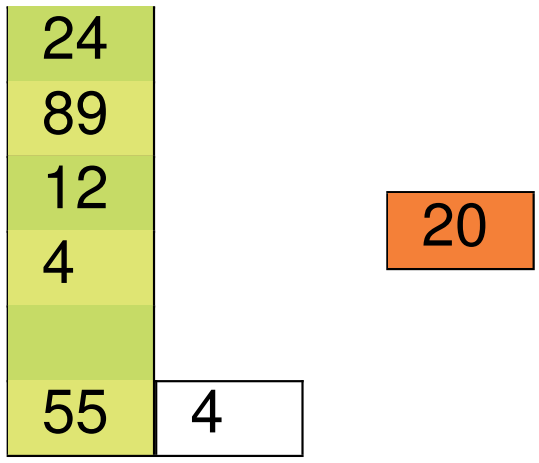
Método `T remove()`

- Testamos se há elementos;
- Decrementamos o fim da fila (último);
- Salvamos o primeiro elemento em variável auxiliar;
- Empurramos tudo para a frente.



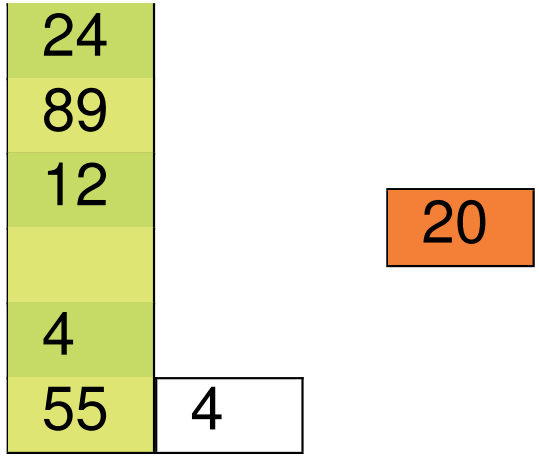
Método `T remove()`

- Testamos se há elementos;
- Decrementamos o fim da fila (último);
- Salvamos o primeiro elemento em variável auxiliar;
- Empurramos tudo para a frente.



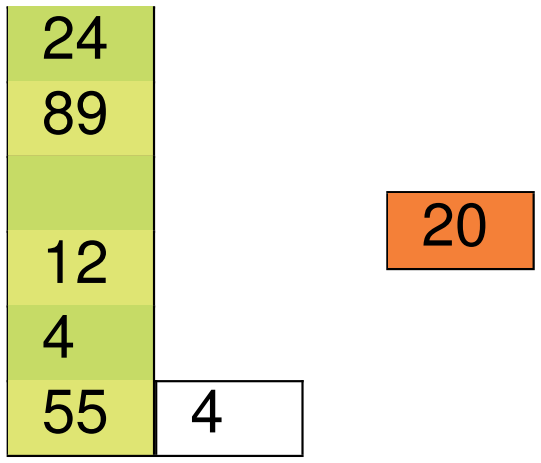
Método `T remove()`

- Testamos se há elementos;
- Decrementamos o fim da fila (último);
- Salvamos o primeiro elemento em variável auxiliar;
- Empurramos tudo para a frente.



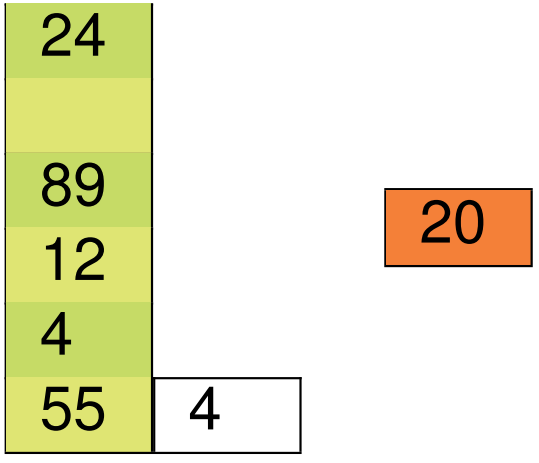
Método `T remove()`

- Testamos se há elementos;
- Decrementamos o fim da fila (último);
- Salvamos o primeiro elemento em variável auxiliar;
- Empurramos tudo para a frente.



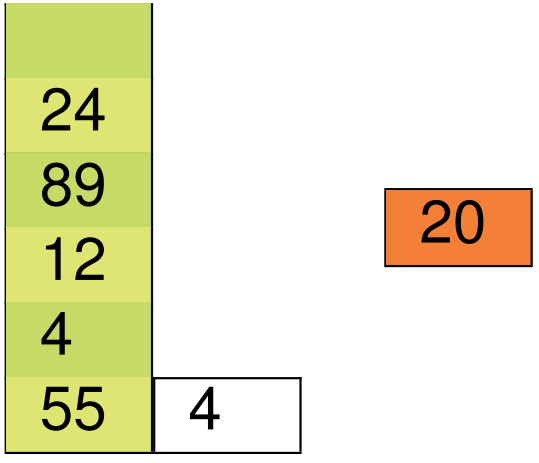
Método `T remove()`

- Testamos se há elementos;
- Decrementamos o fim da fila (último);
- Salvamos o primeiro elemento em variável auxiliar;
- Empurramos tudo para a frente.



Método `T remove()`

- Testamos se há elementos;
- Decrementamos o fim da fila (último);
- Salvamos o primeiro elemento em variável auxiliar;
- Empurramos tudo para a frente.

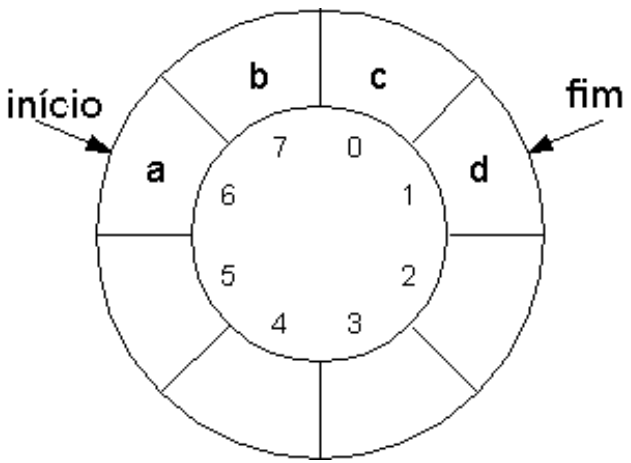


Modelagem eficiente da Fila

- Fila estática circular
  - Não há movimentação de dados;
  - É necessário o registro de `_quant` – **quantidade atual** de elementos;
  - Além do `_fim`, é preciso manter a posição do `_ini` (início) da fila;
  - As atualizações do início e fim devem ser seguidas por uma operação de **divisão inteira**.

Modelagem eficiente da Fila

- Fila estática circular – exemplo de estado



Modelagem eficiente da Fila

- Fila estática circular – **construtor**

```
FilaCircular()
inicio
    T* _dados ← new T[_tam];
    inteiro _quant ← 0;
    inteiro _ini ← 0;
    inteiro _fim ← -1;
fim
```

Modelagem eficiente da Fila

- Fila estática circular – **teste de cheia**

```
bool filaCircularCheia()
inicio
    SE (_quant = _tam) ENTAO
        RETORNE(Verdadeiro);
    SENAO
        RETORNE(Falso);
fim
```

Modelagem eficiente da Fila

- Fila estática circular – **inserção**

```
insereFilaCircular(T dado)
inicio
    SE (filaCircularCheia) ENTAO
        JogueExcecao(ERROFILACHEIA);
    SENAO
        _fim ← ( _fim + 1 )  mod  _tam;
        _dados[_fim] ← dado;
        _quant ← _quant + 1;
    FIM SE
fim
```

Modelagem eficiente da Fila

- Fila estática circular – **remoção**

```
removeFilaCircular()
inicio
    SE (filaCircularVazia) ENTAO
        JogueExcecao(ERROFILAVAZIA);
    SENAO
        dado ← _dados[_ini];
        _ini ← ( _ini + 1 )  mod  _tam;
        _quant ← _quant - 1;
        RETORNE(dado);
    FIM SE
fim
```

Trabalho de Fila em vetor

- Implemente uma classe Fila todas as operações vistas;
- Implemente a fila usando *templates*;
- Implemente a fila com um número de elementos variável definido na instanciação;
- Use as melhores práticas de orientação a objetos;
- Documente todas as classes, métodos e atributos;
- Aplique os testes unitários disponíveis no Moodle da disciplina para validar sua estrutura de dados;
- Entregue até a data definida no Moodle.

Perguntas?


ine
Informática
Estatística
CTC • UFSC


ARS ET SCIENTIA
UNIVERSIDADE FEDERAL
DE SANTA CATARINA
UFSC