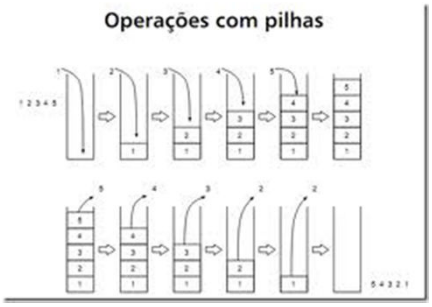
Estrutura de Dados

Rafael D. Ribeiro, M.Sc.
rafaeldiasribeiro@gmail.com
http://www.rafaeldiasribeiro.com.br

Estrutura de Dados

Operações de acesso, inserção e remoção, restrita a estremos da lista. Casos especiais:

- Pilha:
  - Lista linear onde todas as inserções, remoções e acessos são realizados em um único extremo.
  - Lista LIFO (Last In / First Out)



Estrutura de Dados

- Pilha:

- Na alocação sequencial de listas genéricas, considera-se sempre a primeira posição da lista no endereço 1 da memória disponível. Uma alternativa a essa estratégia é a utilização de indicadores especiais, denominados **ponteiros**, para o acesso as posições selecionadas.

- No caso da pilha apenas um ponteiro precisa ser considerado denominado "topo" já que as inserções e remoções são realizadas na mesma extremidade da lista.

Estrutura de Dados

- Pilha (alocação sequencial):

algoritmo 2.7: Inserção na pilha $\mathcal{P}$
se $topo \neq M$ então
    $topo := topo + 1$
    $\mathcal{P}[topo] := novo\text{-}valor$
senão $overflow$

Estrutura de Dados

- Pilha (alocação sequencial):

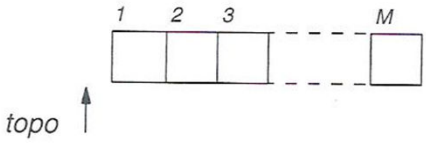
algoritmo 2.8: Remoção da pilha $\mathcal{P}$
se $topo \neq 0$ então
    $valor\text{-}recuperado := \mathcal{P}[topo]$
    $topo := topo - 1$
senão $underflow$

Estrutura de Dados

- Pilha (alocação sequencial):

situação 1:

inicial : pilha vazia



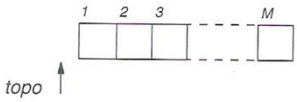
algoritmo 2.7: Inserção na pilha $\mathcal{P}$
se $topo \neq M$ então
    $topo := topo + 1$
    $\mathcal{P}[topo] := novo\text{-}valor$
senão $overflow$

Estrutura de Dados

- Pilha (alocação sequencial):

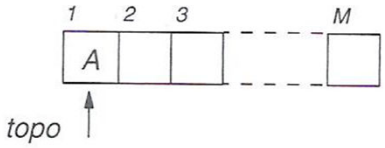
situação 1:

inicial : pilha vazia



situação 2:

inserir informação A



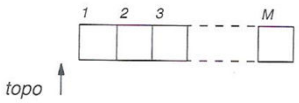
algoritmo 2.7: Inserção na pilha $\mathcal{P}$
se $topo \neq M$ então
$topo := topo + 1$
$\mathcal{P}[topo] := novo\text{-}valor$
senão $overflow$

Estrutura de Dados

- Pilha (alocação sequencial):

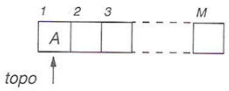
situação 1:

inicial : pilha vazia



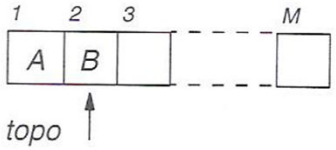
situação 2:

inserir informação A



situação 3:

inserir informação B



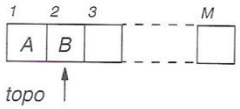
algoritmo 2.7: Inserção na pilha $\mathcal{P}$
se $topo \neq M$ então
$topo := topo + 1$
$\mathcal{P}[topo] := novo\text{-}valor$
senão $overflow$

Estrutura de Dados

- Pilha (alocação sequencial):

situação 3:

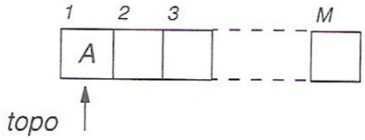
inserir informação B



algoritmo 2.8: Remoção da pilha $\mathcal{P}$
se $topo \neq 0$ então
    $valor\text{-}recuperado := \mathcal{P}[topo]$
    $topo := topo - 1$
senão $underflow$

situação 4:

retirar informação (B)

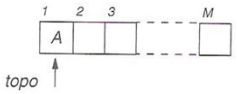


Estrutura de Dados

- Pilha (alocação sequencial):

situação 4:

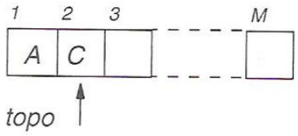
retirar informação (B)



algoritmo 2.7: Inserção na pilha $\mathcal{P}$
se $topo \neq M$ então
    $topo := topo + 1$
    $\mathcal{P}[topo] := novo\text{-}valor$
senão $overflow$

situação 5:

inserir informação C

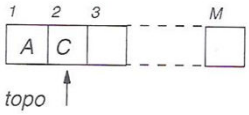


Estrutura de Dados

- Pilha (alocação sequencial):

situação 5:

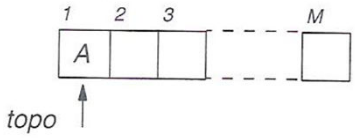
inserir informação C



algoritmo 2.8: Remoção da pilha $\mathcal{P}$
se $topo \neq 0$ então
    $valor\text{-}recuperado := \mathcal{P}[topo]$
    $topo := topo - 1$
senão $underflow$

situação 6:

retirar informação (C)

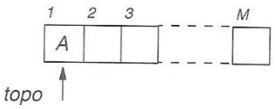


Estrutura de Dados

- Pilha (alocação sequencial):

situação 6:

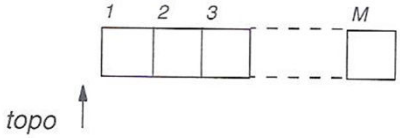
retirar informação (C)



algoritmo 2.8: Remoção da pilha $\mathcal{P}$
se $topo \neq 0$ então
    $valor\text{-}recuperado := \mathcal{P}[topo]$
    $topo := topo - 1$
senão $underflow$

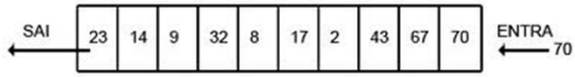
situação 7:

retirar informação (A)



Estrutura de Dados

- Operações de acesso, inserção e remoção, restrita a estremos da lista. Casos especiais:
  - Fila:
    - Lista linear onde todas as inserções são feitas em um certo extremo e todas as remoções e acessos são feitas no outro extremo.
    - Lista FIFO (First In / First Out)



Estrutura de Dados

- As filas precisam de uma implementação um pouco mais elaborada. São necessários dois ponteiros: inicio da fila (**f**) e retaguarda (**r**).
- Para adição de um novo elemento move-se (**r**) para retirada (**f**). A situação da fila vazia é representada **f = r = 0**
- A medida que que os ponteiros são incrementados na memória disponível, a fila "se move", o que pode dar à falsa impressão de memória esgotada. Para evitar este problema, consideram-se os M nós alocados como se estivessem em círculo, onde **F[1]** segue **F[M]**.
- No algoritmo de inserção , a variável ***prov*** armazena provisoriamente a posição de memória calculada de forma a respeitar a circularidade, só sendo movimentado o ponteiro **r** se a operação for possível.

Estrutura de Dados

- Fila (alocação sequencial)

```
algoritmo 2.9: Inserção na fila F
prov := r mod M + 1
se prov ≠ f então
        r := prov
        F[r] := novo-valor
        se f = 0 então
                f := 1
senão overflow
```

Estrutura de Dados

- Fila (alocação sequencial)

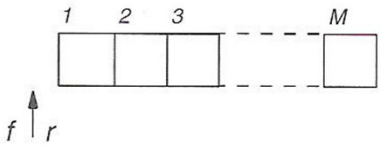
```
algoritmo 2.10: Remoção da fila F
se  f ≠ 0 então
        valor-recuperado := F[f]
        se f = r então
                f := r := 0
        senão f := f mod M + 1
senão underflow
```

Estrutura de Dados

- Fila (alocação sequencial)

situação 1:

inicial : fila vazia



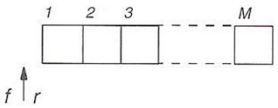

```
algoritmo 2.9: Inserção na fila F
prov := r mod M + 1
se prov ≠ f então
        r := prov
        F[r] := novo-valor
        se f = 0 então
                f := 1
senão overflow
```

Estrutura de Dados

- Fila (alocação sequencial)

situação 1:

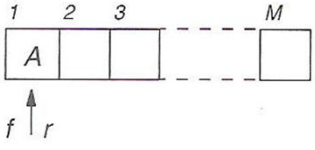
inicial : fila vazia




```
algoritmo 2.9: Inserção na fila F
prov := r mod M + 1
se prov ≠ f então
        r := prov
        F[r] := novo-valor
        se f = 0 então
                f := 1
senão overflow
```

situação 2:

inserir informação A

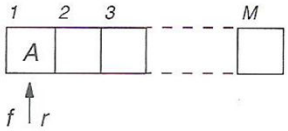


Estrutura de Dados

- Fila (alocação sequencial)

situação 2:

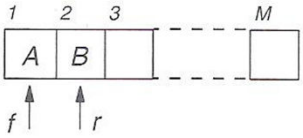
inserir informação A




```
algoritmo 2.9: Inserção na fila F
prov := r mod M + 1
se prov ≠ f então
        r := prov
        F[r] := novo-valor
        se f = 0 então
                f := 1
senão overflow
```

situação 3:

inserir informação B

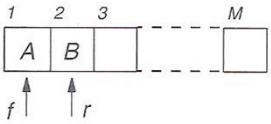


Estrutura de Dados

- Fila (alocação sequencial)

situação 3:

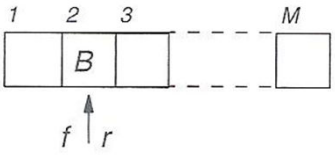
inserir informação B




```
algoritmo 2.10: Remoção da fila F
se  f ≠ 0 então
        valor-recuperado := F[f]
        se f = r então
                f := r := 0
        senão f := f mod M + 1
senão underflow
```

situação 4:

retirar informação (A)

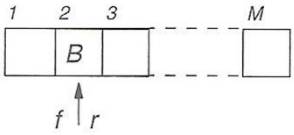


Estrutura de Dados

- Fila (alocação sequencial)

situação 4:

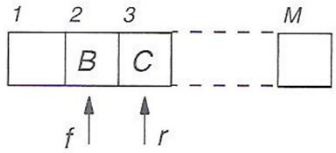
retirar informação (A)




```
algoritmo 2.10: Remoção da fila F
se  f ≠ 0 então
      valor-recuperado := F[f]
      se f = r então
            f := r := 0
      senão f := f mod M + 1
senão underflow
```

situação 5:

inserir informação C

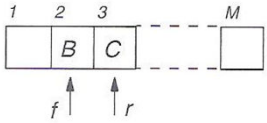


Estrutura de Dados

- Fila (alocação sequencial)

situação 5:

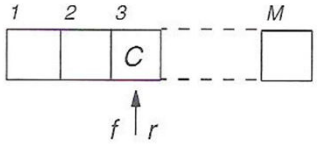
inserir informação C




```
algoritmo 2.9: Inserção na fila F
prov := r mod M + 1
se prov ≠ f então
      r := prov
      F[r] := novo-valor
      se f = 0 então
            f := 1
senão overflow
```

situação 6:

retirar informação (B)

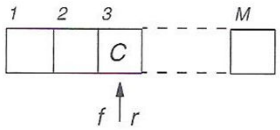


Estrutura de Dados

- Fila (alocação sequencial)

situação 6:

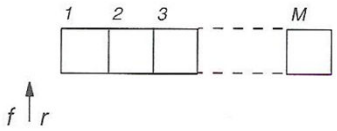
retirar informação (B)




```
algoritmo 2.10: Remoção da fila F
se  f ≠ 0 então
      valor-recuperado := F[f]
      se f = r então
            f := r := 0
      senão f := f mod M + 1
senão underflow
```

situação 7:

retirar informação (C)



Estrutura de Dados

Bibliografia das Notas de Aula

Estruturas de Dados e Seus Algoritmos - Jayme Luiz Szwarcfiter, Lilian Markenzon LTC Editora - 2 ° Ed.