

# ACH2023 - Algoritmos e Estruturas de Dados I

Pilhas e filas

Prof. Flávio Luiz Coutinho  
flcoutinho@usp.br

# Pilhas e Filas

Assim como as listas, são estruturas lineares:

- organização lógica sequencial

# Pilhas e Filas

Assim como as listas, são estruturas lineares:

- organização lógica sequencial

Mas possuem regras mais rígidas quanto à entrada e saída dos elementos.

# Pilhas e Filas

Nas implementações de lista (em suas várias versões):

- especificamos a posição de inserção de um elemento  
ou
- o elemento é inserido de modo a manter a lista ordenada por valor  
e, além disso,
- solicitamos a remoção de um elemento específico (poderíamos pensar também em uma variação da remoção, que remove um elemento de uma posição específica).

# Pilhas e Filas

Em uma pilha:

- a inserção de um novo elemento ocorre sempre no topo da pilha.
- a operação de remoção remove (e devolve) sempre o elemento que reside no topo da pilha.

Funciona de forma análoga a uma pilha (de papel, documentos, etc) do mundo real.

# Pilhas e Filas

Em uma fila:

- a inserção de um novo elemento ocorre sempre no final da fila.
- a operação de remoção remove (e devolve) sempre o elemento que ocupa a primeira posição da fila.

Funciona de forma análoga a uma fila (do banco, supermercado, etc) do mundo real.

# Pilhas e Filas

Pilhas e filas são, portanto, especializações de listas lineares com propósitos mais específicos.

E podemos usar as implementações de lista que já estudamos como base para implementar pilhas e filas.

# Pilha implementada a partir de uma lista

Operação push (empilha):

```
insere(lista, x, tamanho(lista))
```

Operação pop (desempilha):

```
removido = elemento_na_posicao(lista, tamanho(lista) - 1)
```

```
remove_elemento_na_posicao(lista, tamanho(lista) - 1)
```



# Fila implementada a partir de uma lista

Operação adiciona\_na\_fila:

```
insere(lista, x, tamanho(lista))
```

Operação remove\_da\_fila:

```
removido = elemento_na_posicao(lista, 0)
```

```
remove_elemento_na_posicao(lista, 0)
```

# Pilha/fila implementada a partir de uma lista

Nossas implementações base de lista serão sempre uma boa base para implementar uma pilha e/ou fila?

Precisamos tomar um pouco de cuidado...

- lista sequencial como base de pilha / fila
- lista ligada como base de pilha / fila
- quais as implicações de se usar cada tipo de lista como base?

# Lista sequencial como base da implementação

Adequada para **pilha**:

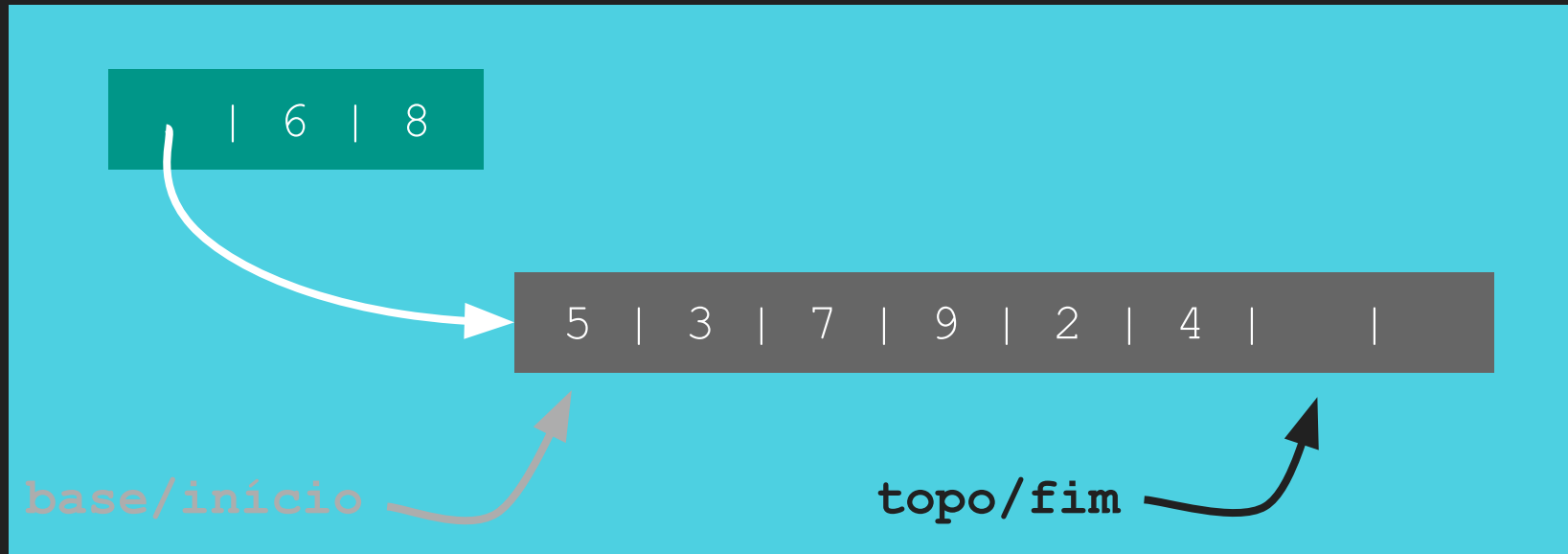
- inserções e remoções do final eficientes (tempo constante).

Ruim para **fila**:

- inserções no final são eficientes, mas remoções do início não são (exigem deslocamento de todos os elementos restantes para a esquerda).

# Lista sequencial como base da implementação

Pilha e fila baseada em uma lista sequencial:



# Lista sequencial como base da implementação

Adequada para **pilha**:

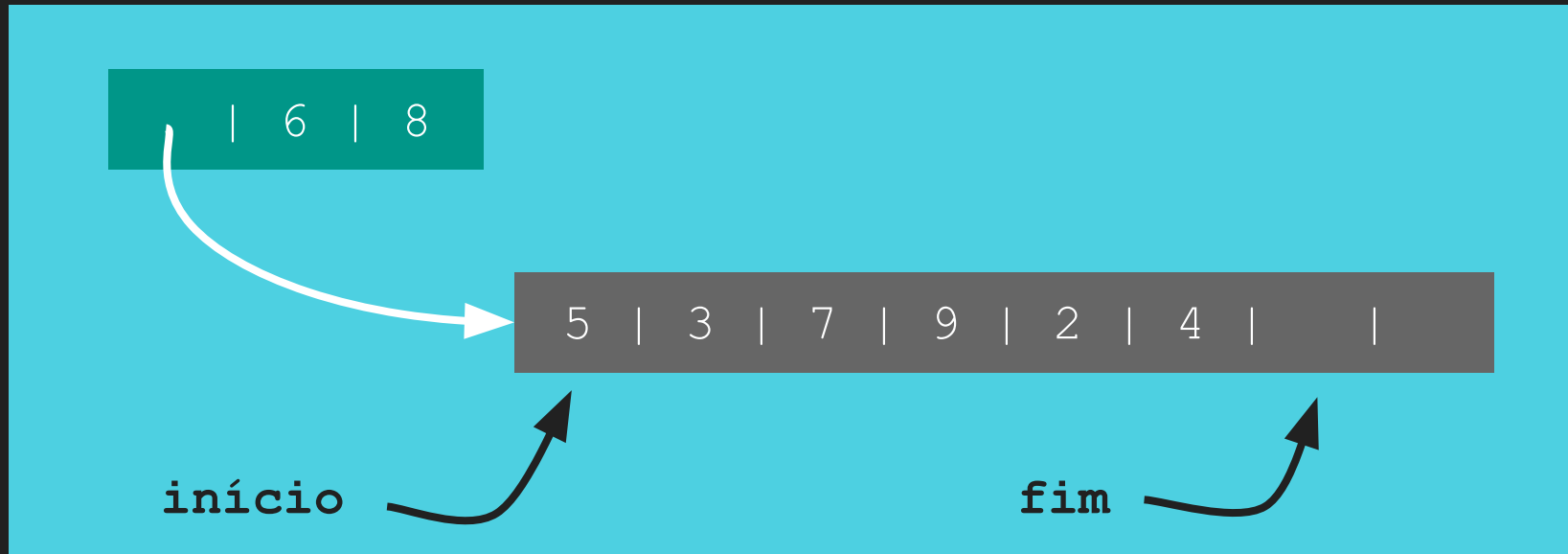
- inserções e remoções do final eficientes (tempo constante).

Ruim para **fila**:

- inserções no final são eficientes, mas remoções do início não são (exigem deslocamento de todos os elementos restantes para a esquerda).
- Felizmente, é fácil contornar esta limitação: basta manter, na estrutura, um campo adicional que guarda um índice indicando o início da lista.

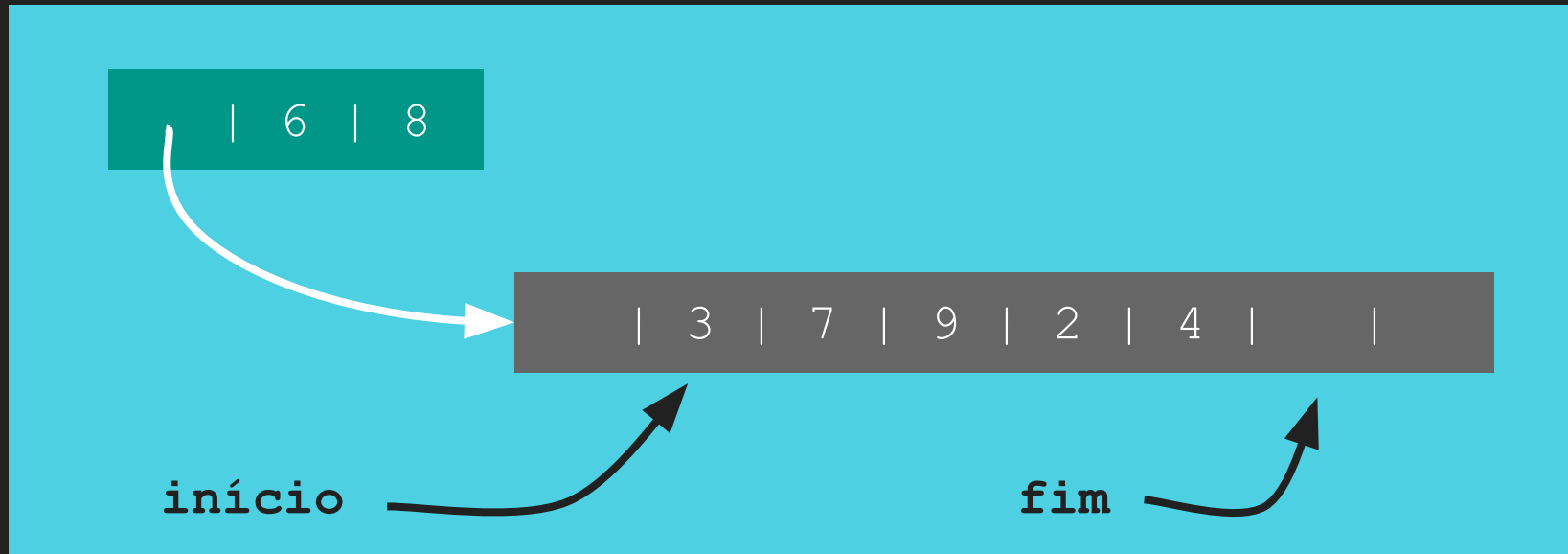
# Lista sequencial como base da implementação

Fila baseada em uma lista sequencial:



# Lista sequencial como base da implementação

Fila baseada em uma lista sequencial:



# Lista sequencial como base da implementação

Fila baseada em uma lista sequencial:





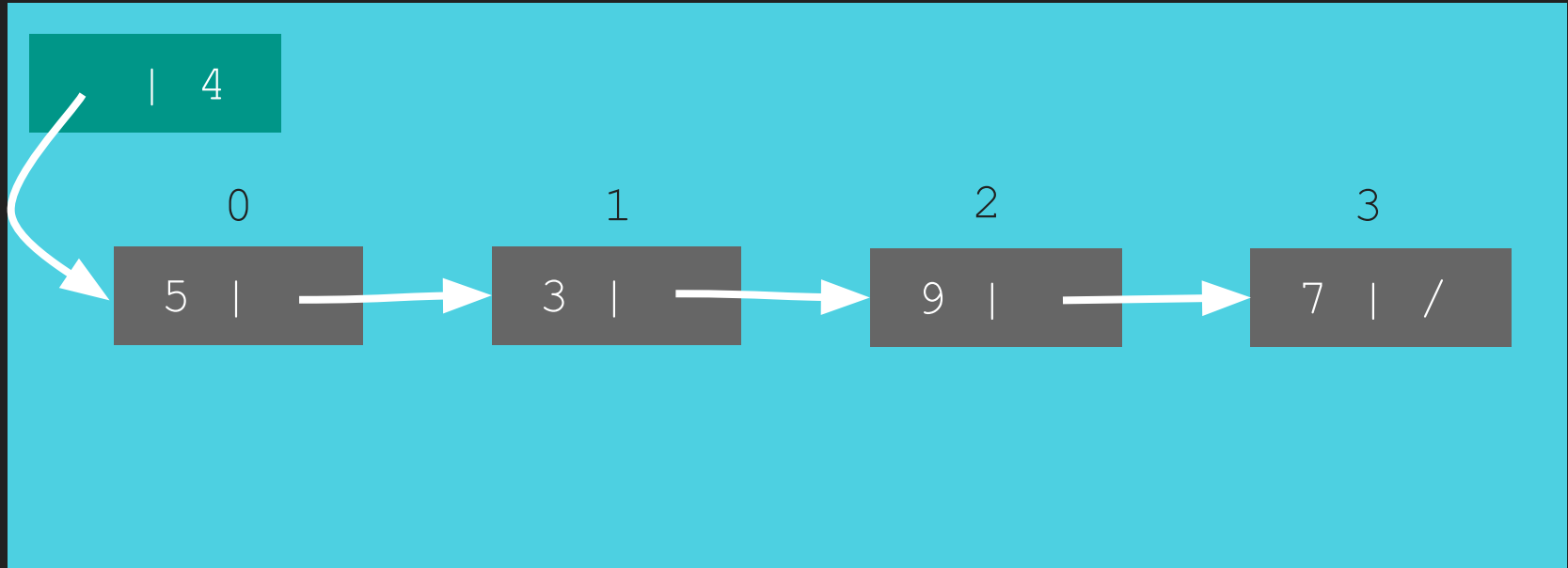
# Lista Ligada como base da implementação

À primeira vista, não é adequada para **pilha**:

- inserções e remoções do final da fila são custosas (custo proporcional a  $n$ ).

# Lista Ligada como base da implementação

Pilha baseada em lista ligada:



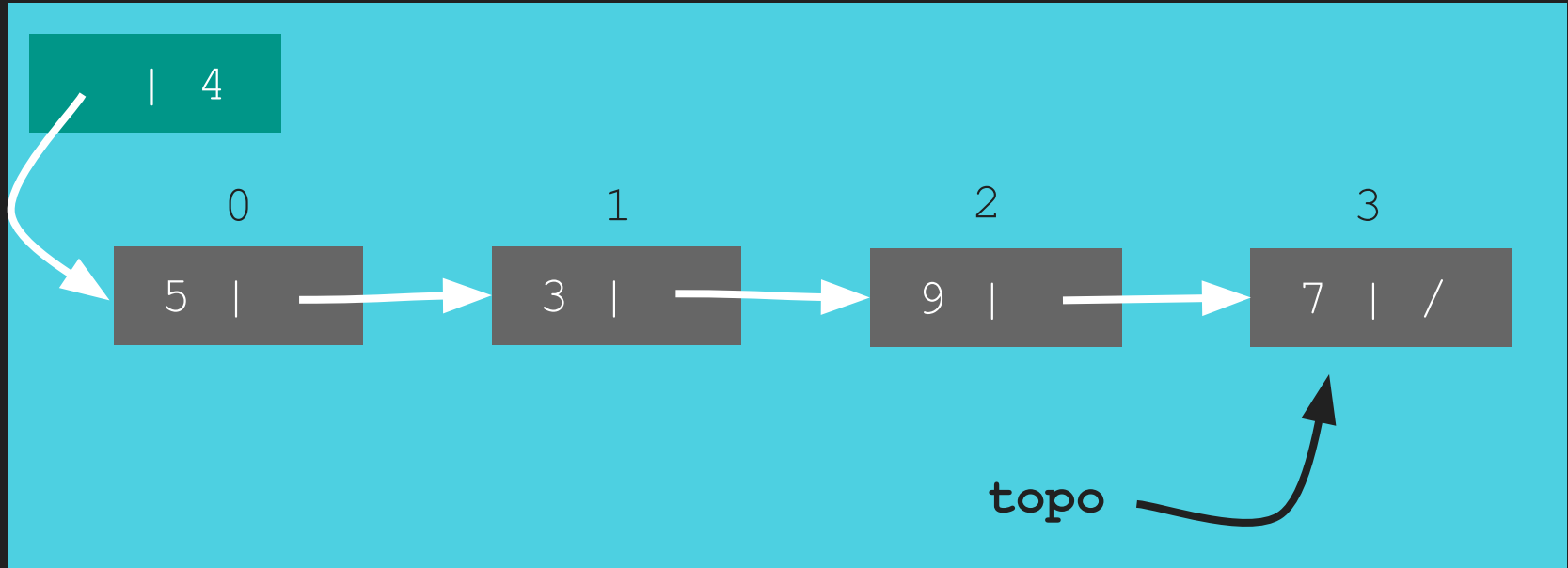
# Lista Ligada como base da implementação

À primeira vista, não é adequada para **pilha**:

- inserções e remoções do final da fila são custosas (custo proporcional a  $n$ ).
- Seria uma solução alterar a estrutura da lista ligada para que ela mantenha um ponteiro também para o último nó da lista?

# Lista Ligada como base da implementação

Pilha baseada em lista ligada:



# Lista Ligada como base da implementação

À primeira vista, não é adequada para **pilha**:

- inserções e remoções do final da fila são custosas (custo proporcional a  $n$ ).
- Seria uma solução alterar a estrutura da lista ligada para que ela mantenha um ponteiro também para o último nó da lista? **Ajuda na inserção, mas a remoção continua sendo uma operação custosa.**

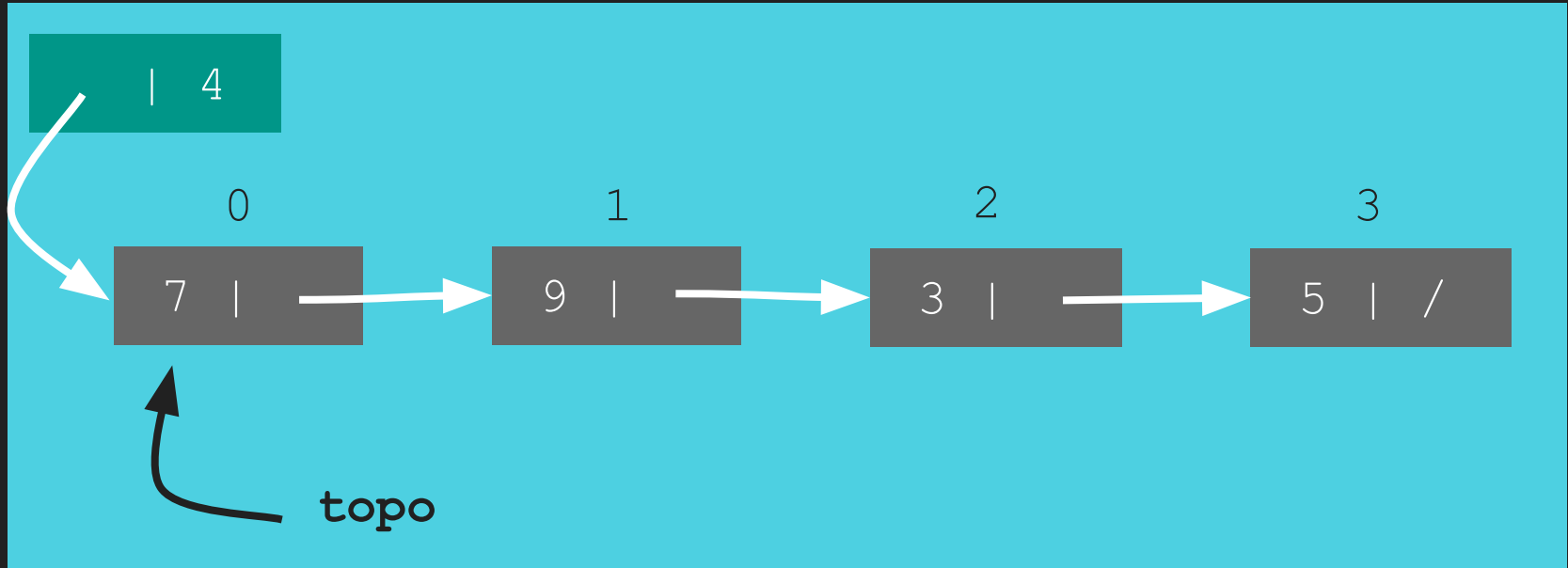
# Lista Ligada como base da implementação

À primeira vista, não é adequada para **pilha**:

- inserções e remoções do final da fila são custosas (custo proporcional a  $n$ ).
- Seria uma solução alterar a estrutura da lista ligada para que ela mantenha um ponteiro também para o último nó da lista? **Ajuda na inserção, mas a remoção continua sendo uma operação custosa.**
- Solução: inserir/remover do início da fila (continua sendo uma pilha).

# Lista Ligada como base da implementação

Pilha baseada em lista ligada:



# Lista Ligada como base da implementação

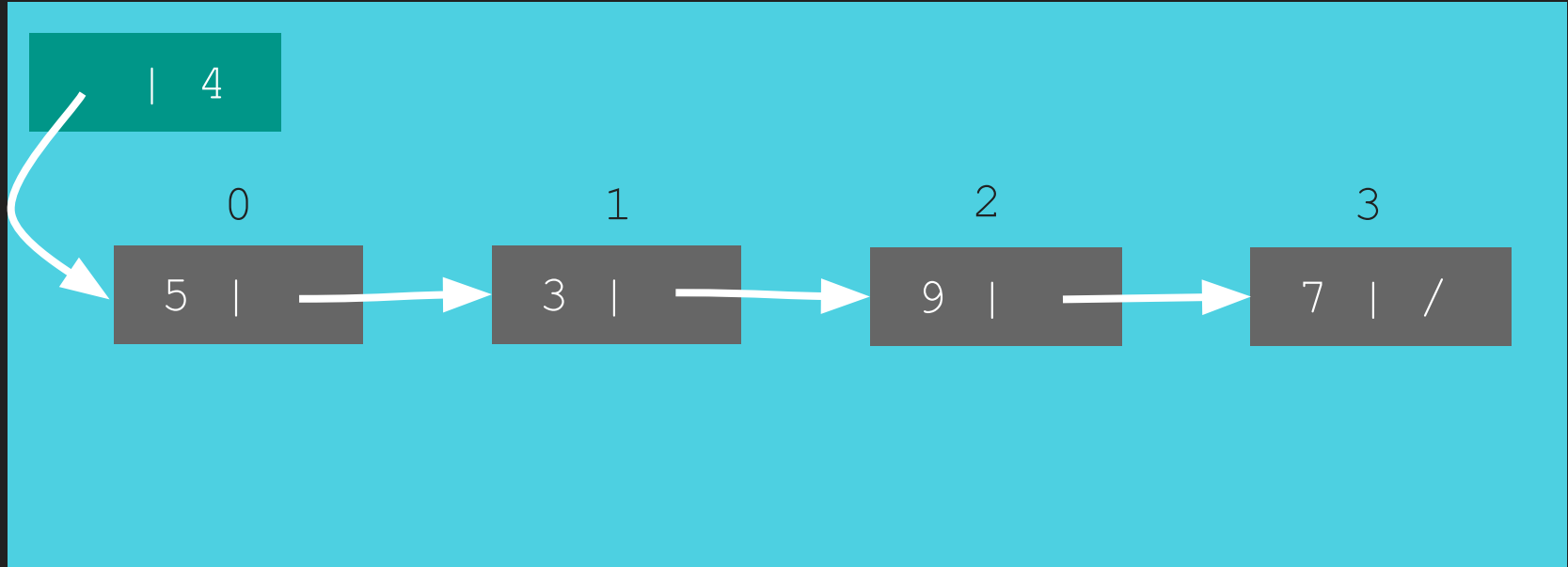
À primeira vista, também não é adequada para implementar uma **fila**:

- remoções do início eficientes, mas inserções no final custosas.



# Lista Ligada como base da implementação

Fila baseada em lista ligada:



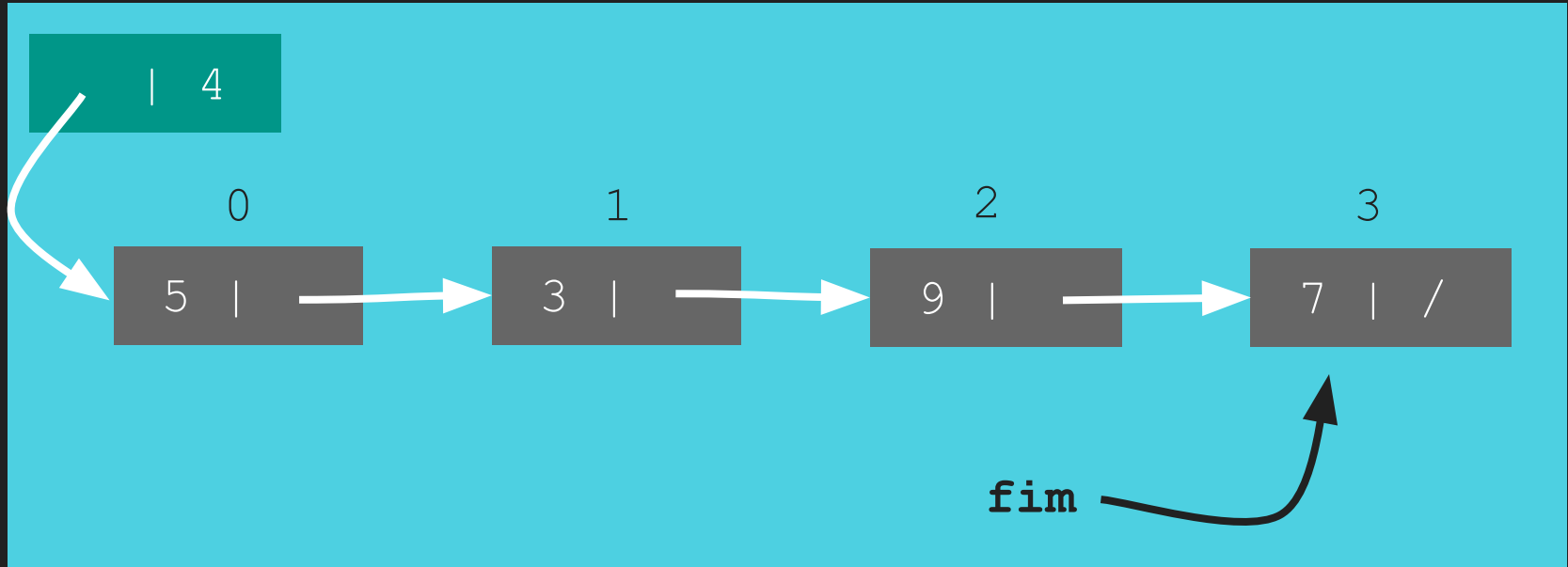
# Lista Ligada como base da implementação

À primeira vista, também não é adequada para implementar uma **fila**:

- remoções do início eficientes, mas inserções no final custosas.
- Seria uma solução alterar a estrutura da lista ligada para que ela mantenha um ponteiro também para o último nó da lista?

# Lista Ligada como base da implementação

Fila baseada em lista ligada:



# Lista Ligada como base da implementação

À primeira vista, também não é adequada para implementar uma **fila**:

- remoções do início eficientes, mas inserções no final custosas.
- Seria uma solução alterar a estrutura da lista ligada para que ela mantenha um ponteiro também para o último nó da lista? No caso da fila seria uma boa ideia. Inserções no final e remoções do início podem ser implementadas de forma eficiente (tempo constante), se a lista ligada mantém ponteiros para o primeiro e último nós.

# Conclusão

É possível implementar as operações de inserção e remoção em pilhas e filas com complexidade de tempo constante, tomando como base de implementação tanto listas sequenciais quanto listas ligadas (embora, adaptações possam, eventualmente, ser necessárias).