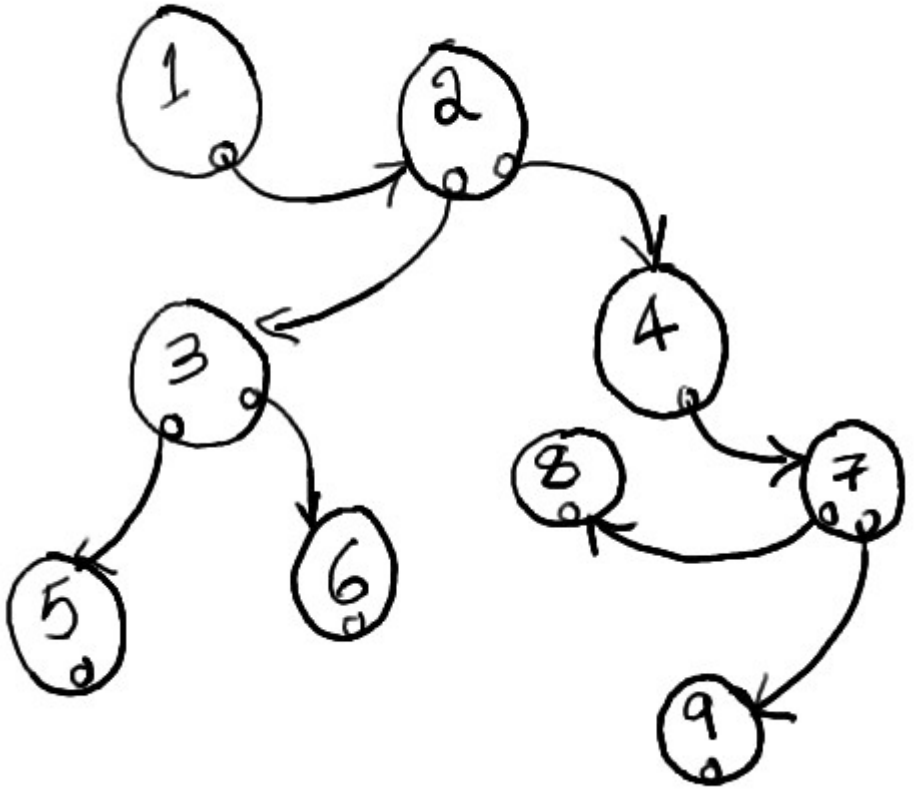
Estruturas de Dados Avançadas

TIPOS ABSTRATOS DE DADOS
Conjuntos dinâmicos

AEDS-II

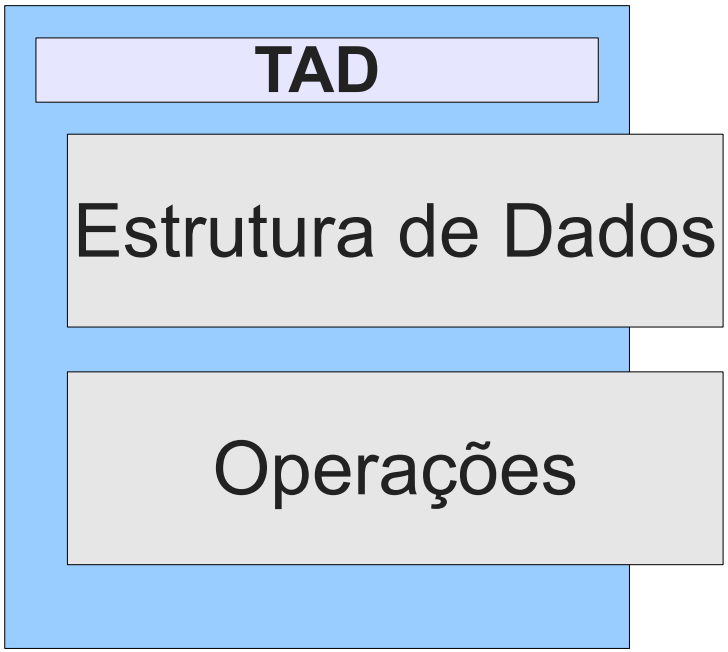
Prof. Renê R. Veloso
Eng. Sistemas
2º Período

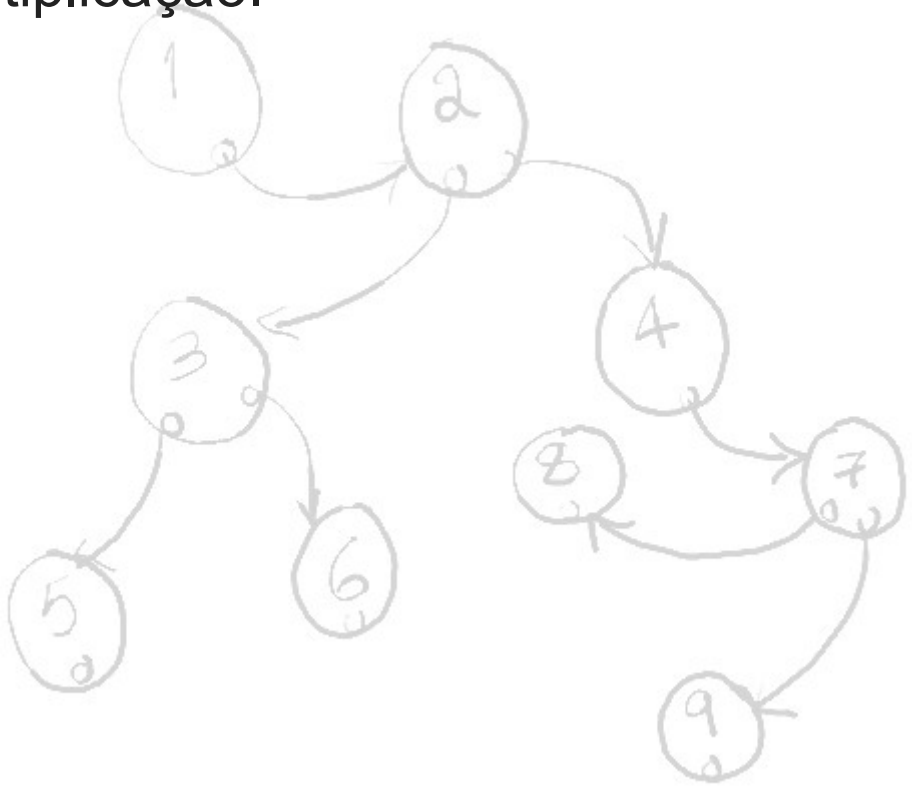


Tipos Abstratos de Dados

TAD: abstrai um conjunto de dados e as operações sobre esse conjunto.

Exemplo: o conjunto dos inteiros acompanhado das operações de adição, subtração e multiplicação.





Tipos Abstratos de Dados

Os conjunto manipulados por algoritmos podem crescer, encolher ou sofrer outras mudanças ao longo do tempo.

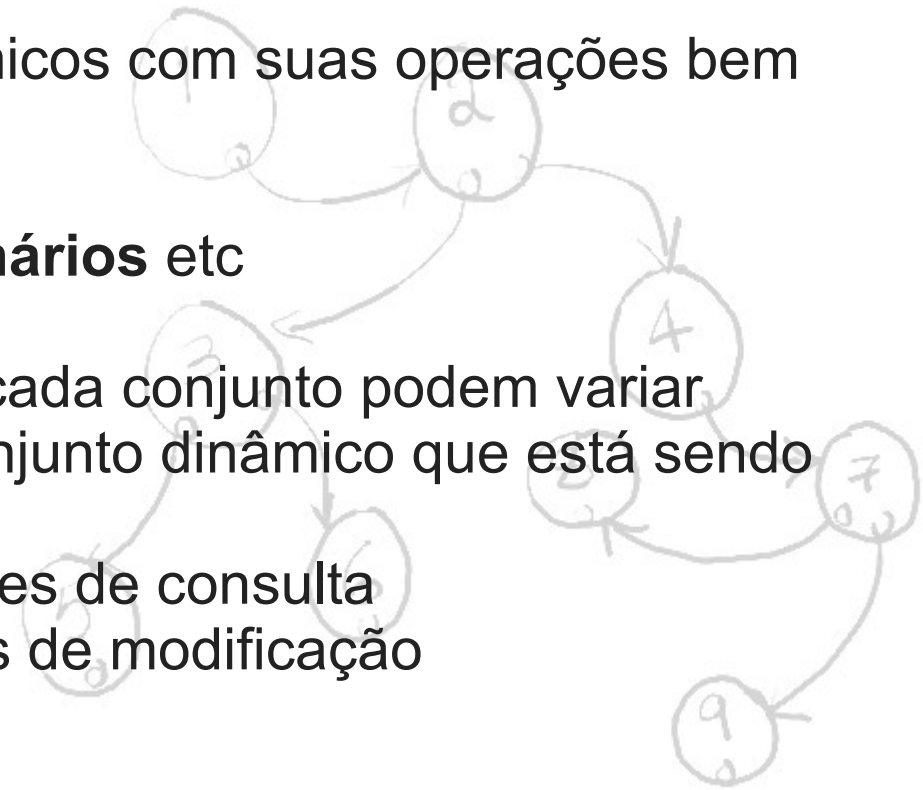
Conjuntos Dinâmicos

Exemplos de conjuntos dinâmicos com suas operações bem definidas são:

Filas, pilhas, árvores, dicionários etc

As operações básicas sobre cada conjunto podem variar definindo, assim, o tipo de conjunto dinâmico que está sendo manipulado.

Operações de consulta
Operações de modificação



Operações Sobre Conjuntos Dinâmicos

As operações típicas são as seguintes:

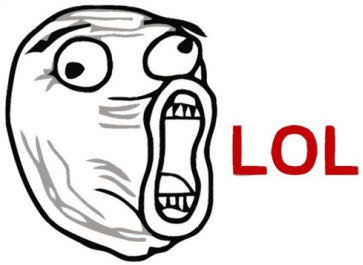
- ***Busca (S, k)*** : dado um conjunto S e um valor de chave k, retorna um apontamento x para um elemento em S tal que S[x]=k, ou NULO se nenhum elemento for encontrado.
- ***Insere (S, x)*** : Operação de modificação que aumenta o conjunto S com o elemento apontado por x.
- ***Remove (S, x)*** : Operação de modificação que, dado um apontamento x para um elemento de S, remove x de S. (Exige um apontamento e não um valor de chave.)
- ***Mínimo (S)*** : Uma consulta sobre um conjunto totalmente ordenado S que retorna um apontamento para o elemento de S com a menor chave.
- ***Máximo (S)*** : Uma consulta sobre um conjunto totalmente ordenado S que retorna um apontamento para o elemento de S com a maior chave.
- ***Sucessor (S, x)*** : Uma consulta que, dado um conjunto S e um apontamento x para um elemento de S, retorna um apontamento para o maior elemento seguinte em S, ou NULO se x é o elemento máximo.
- ***Predecessor (S, x)*** : Uma consulta que, dado um conjunto S e um apontamento x para um elemento de S, retorna um apontamento para o menor elemento seguinte em S, ou NULO se x é o elemento mínimo.

Obs.: Para um conjunto de *n* chaves, uma chamada a **Mínimo** seguida de *n-1* chamadas a **Sucessor** enumera os elementos do conjunto em sequência ordenada.

Estruturas de Dados Elementares

TADs: Pilhas e Filas

… chegou a hora!



Pilhas e Filas

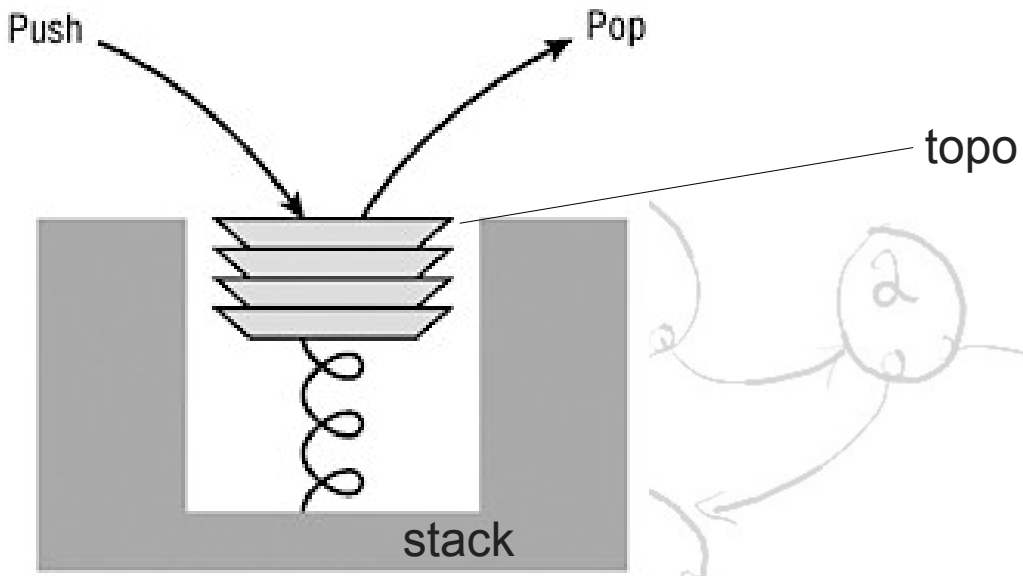
São TADs de conjuntos dinâmicos nos quais o elemento removido do conjunto pela operação REMOVER é especificado previamente.

Pilha: o elemento eliminado do conjunto é o mais recentemente inserido.
Implementa a regra de ***último a entrar, primeiro a sair***,
ou **LIFO** *(last-in, first-out).*

Fila: o elemento eliminado é sempre o que esteve no conjunto pelo tempo mais longo.
Implementa a regra de ***primeiro a entrar, primeiro a sair***,
ou **FIFO** (*first-in, first-out*).

Pilhas

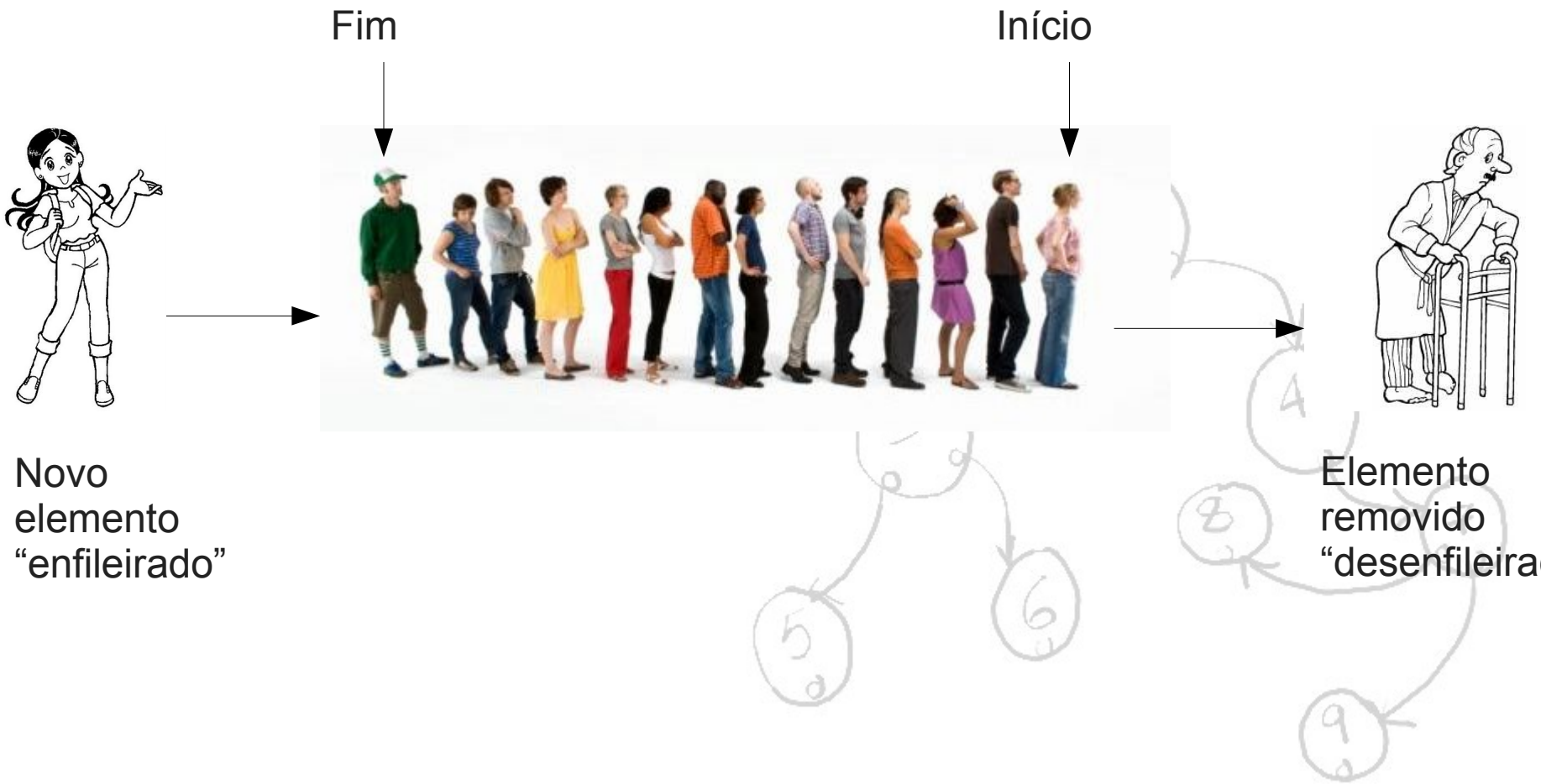
Pilha (ou *stack*): a operação INSERIR em uma pilha é frequentemente chamada de **PUSH**
E a operação de REMOVER é chamada de **POP**



Dessa forma, somente o elemento do **topo** está acessível.
Se não existe elemento (ou seja, NULO) no topo da pilha, dizemos que a pilha está vazia.
Ao tentarmos remover algo de uma pilha vazia, dizemos que houve um ***estouro negativo.***
Se a pilha tiver um tamanho máximo, ao tentar inserir novo elemento, dizemos que houve um ***estouro positivo***.

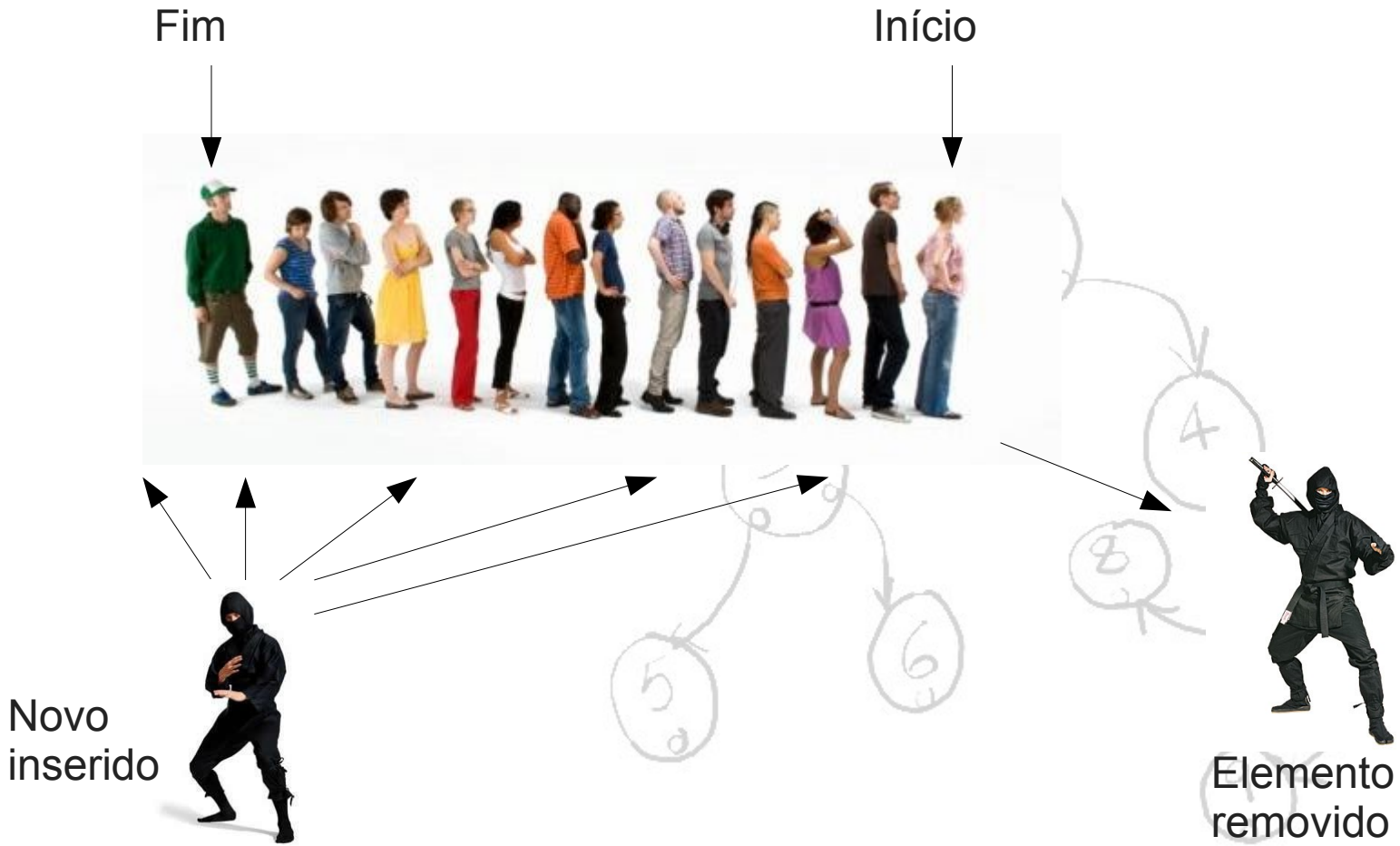
Filas

Fila (ou *Queue*): não possui a noção de topo, mas de **início** e **fim** da fila.



TAD genérico: LISTA

Lista (ou *List*): é um TAD genérico que não possui regras fixas de manipulação. Em tese, sua implementação deve permitir operações de ***inserção e remoção*** de **qualquer posição do conjunto**.



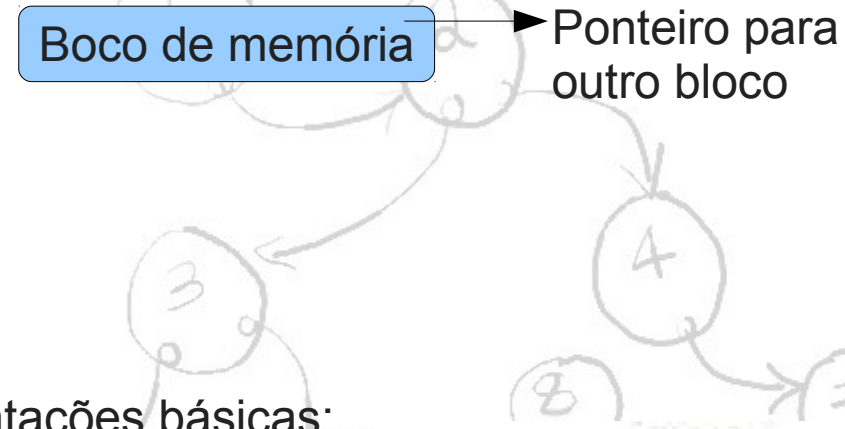
Pilha e Fila: Listas

Quando falamos em implementação, uma pilha e uma fila são codificadas seguindo uma mesma organização de blocos de memória.

Os blocos de memória são conectados formando um encadeamento por ponteiros que permite a navegação entre os elementos.

```
struct bloco {
    //dados
    struct bloco *proximo;
};

typedef struct bloco Bloco;
```

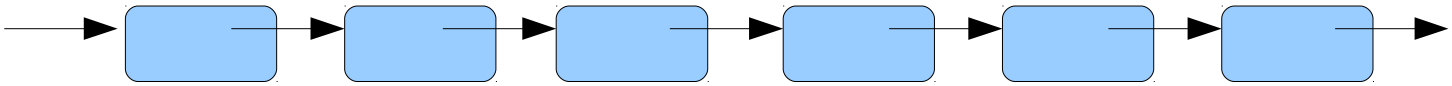


O encadeamento pode seguir três orientações básicas:

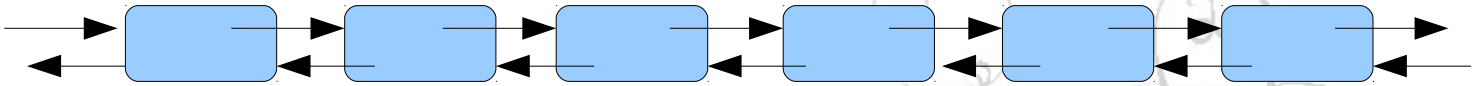
- **Encadeamento simples**
- **Encadeamento duplo**
- **Encadeamento circular (simples ou duplo)**

Encadeamentos

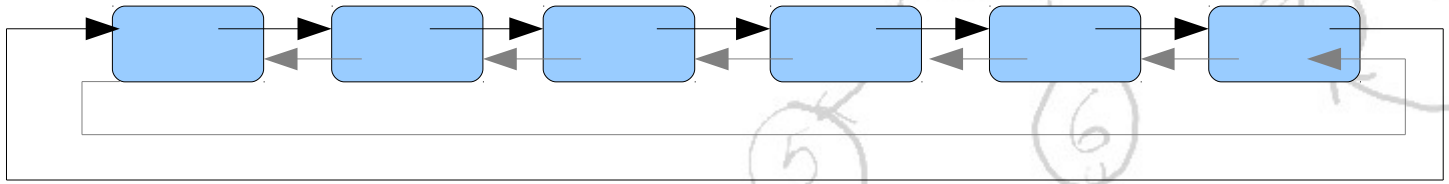
Simples



Duplo



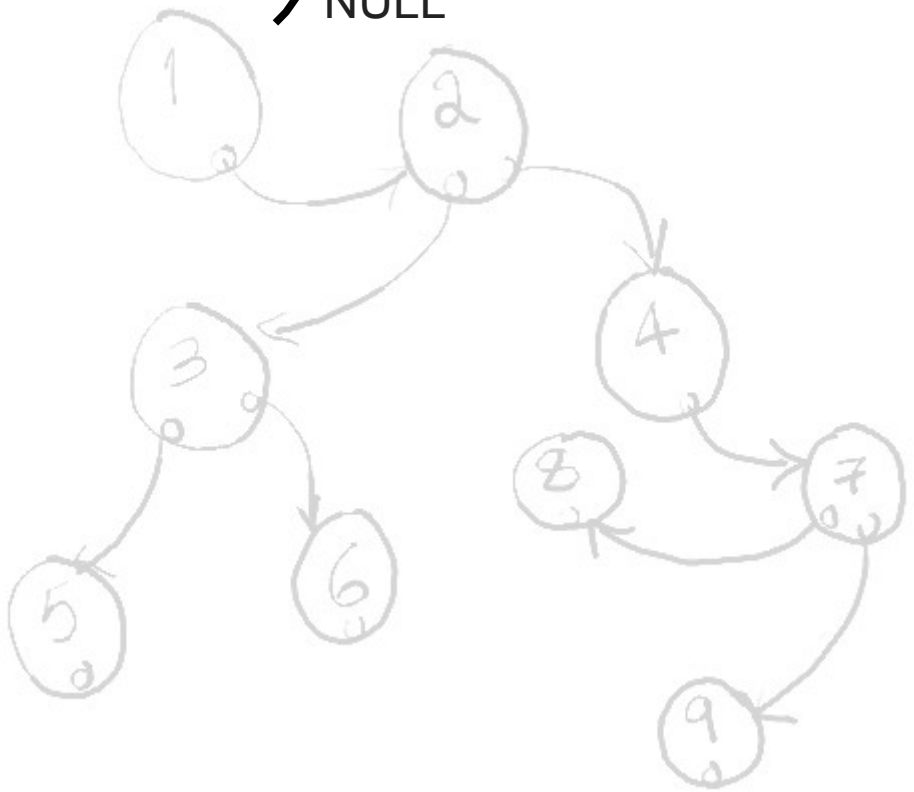
Circular



Dinâmica da Pilha

Elementos a serem inseridos: 1, 2, 3

Configuração inicial da pilha: pilha.***topo*** → NULL

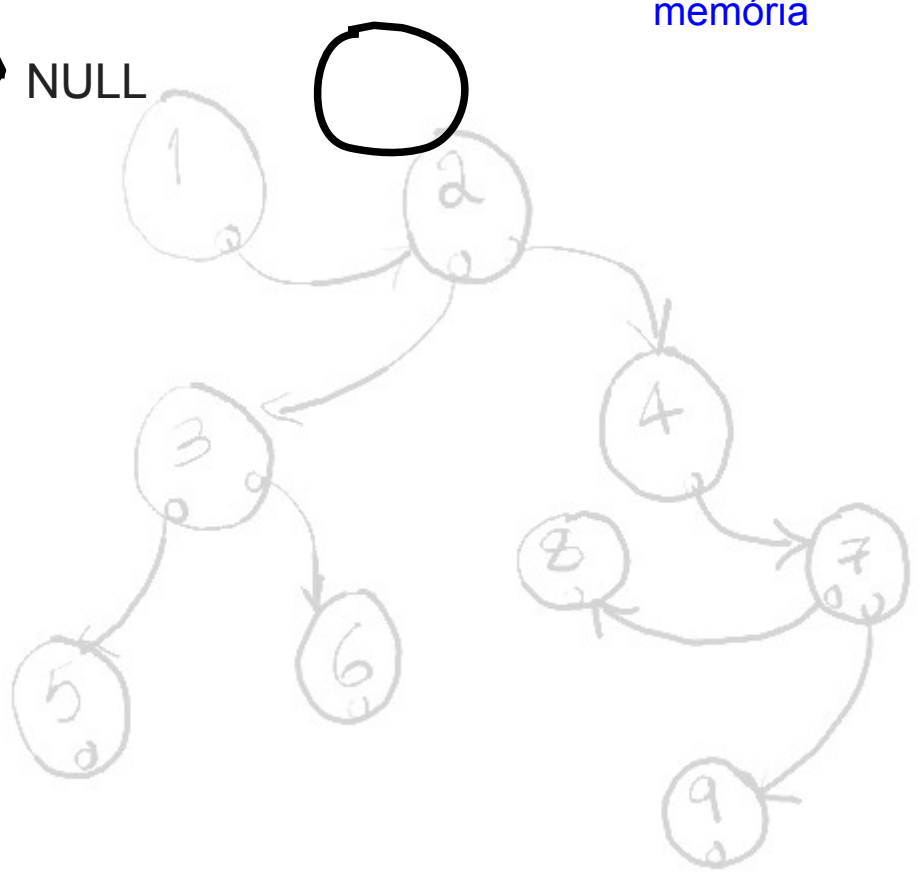


Dinâmica da Pilha

Elementos a serem inseridos: 1, 2, 3

PUSH(pilha,1): pilha.***topo*** → NULL

Aloca bloco de memória

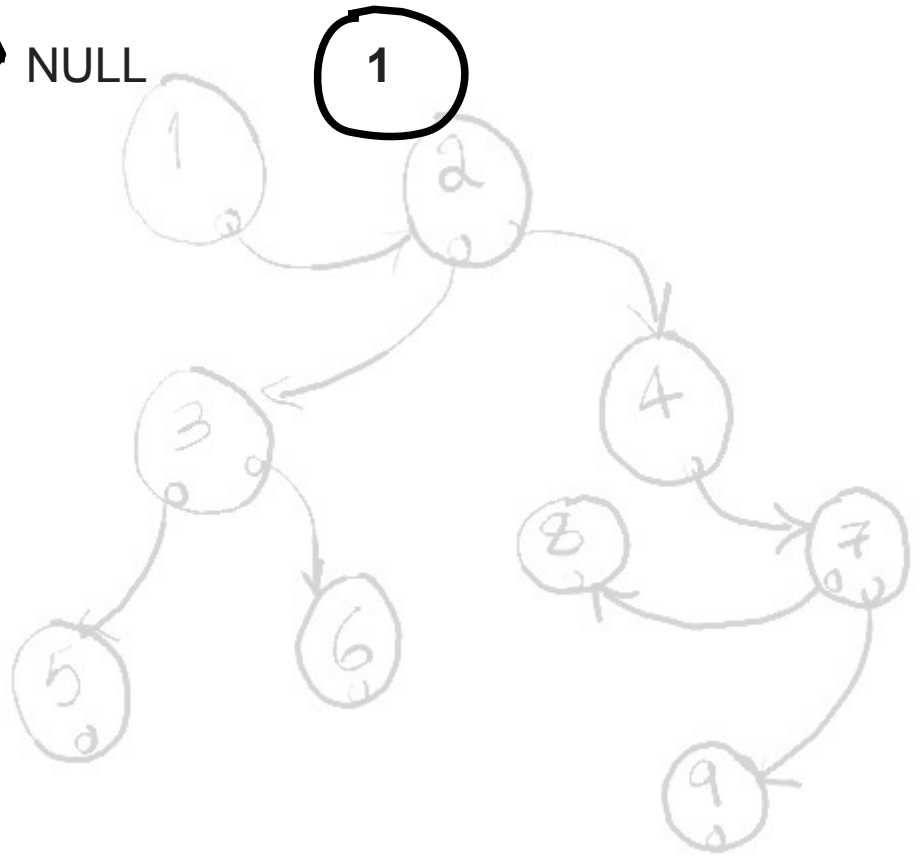


Dinâmica da Pilha

Elementos a serem inseridos: 1, 2, 3

Insere item no bloco

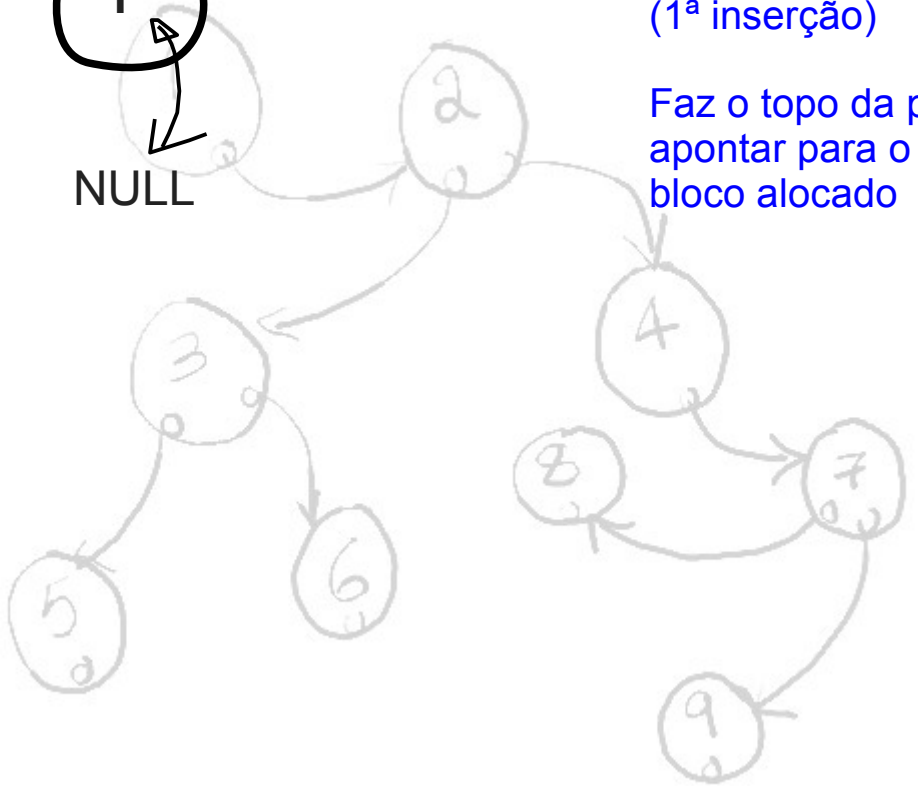
PUSH(pilha,1): pilha.***topo*** → NULL



Dinâmica da Pilha

Elementos a serem inseridos: 1, 2, 3

PUSH(pilha,1): pilha.***topo*** → 1 → NULL



Indica como NULL o próximo elemento do novo bloco (1ª inserção)

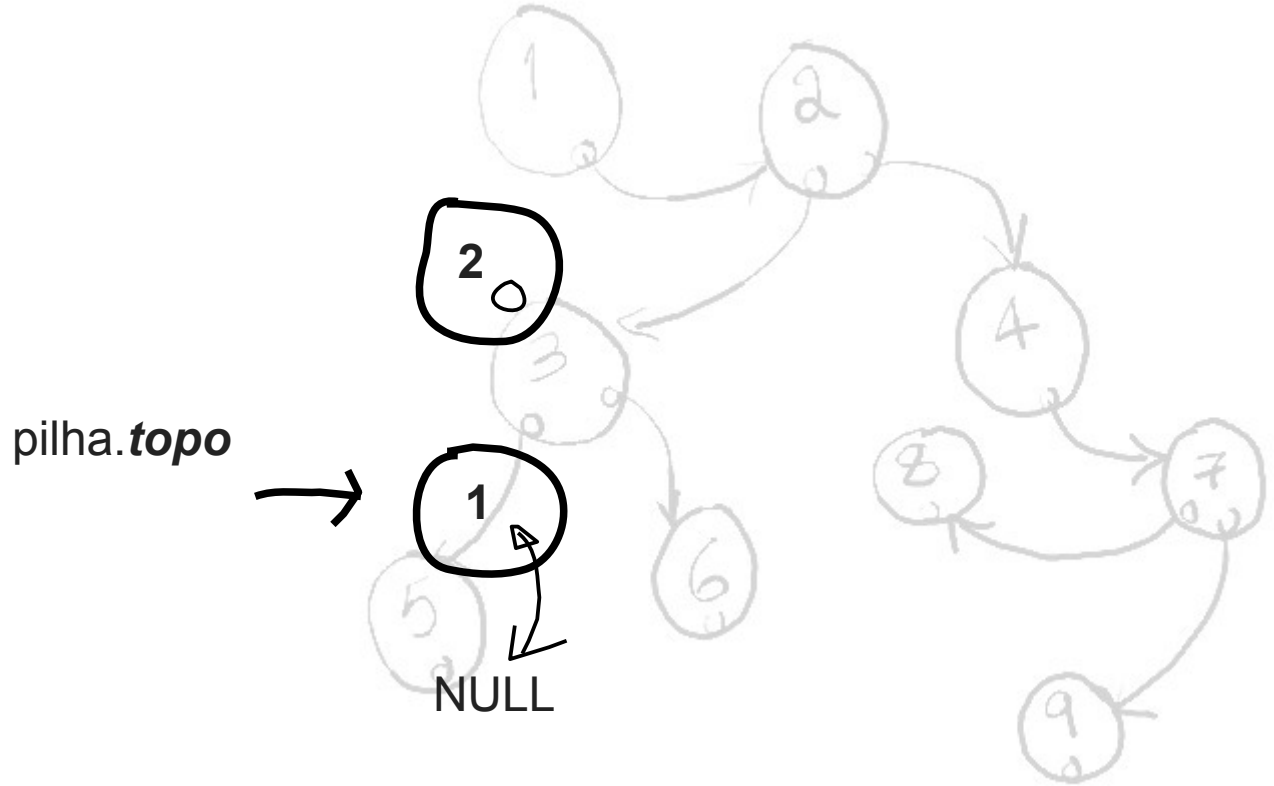
Faz o topo da pilha apontar para o novo bloco alocado

Dinâmica da Pilha

Elementos a serem inseridos: 1, 2, 3

PUSH(pilha,2):

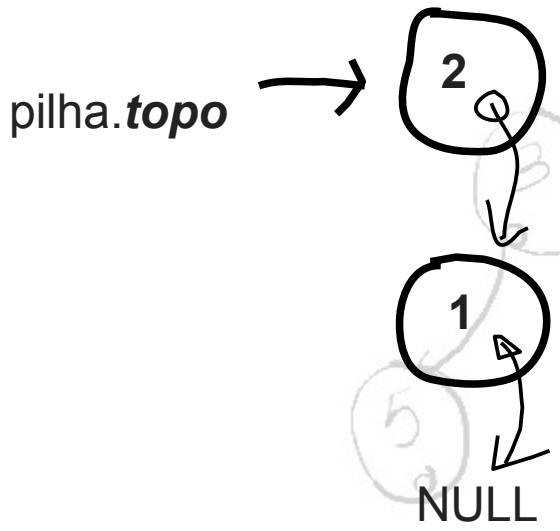
Aloca novo bloco
e insere o elemento
nele



Dinâmica da Pilha

Elementos a serem inseridos: 1, 2, 3

PUSH(pilha,2):



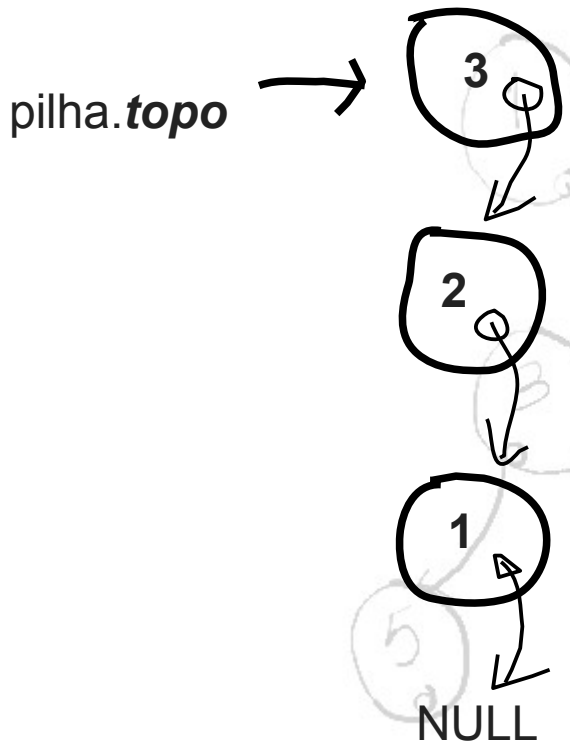
Faz o ponteiro de *próximo* apontar para o topo da pilha

Faz o topo da pilha apontar para o novo bloco

Dinâmica da Pilha

Elementos a serem inseridos: 1, 2, 3

PUSH(pilha,3):



Cria novo bloco

Insere elemento nele

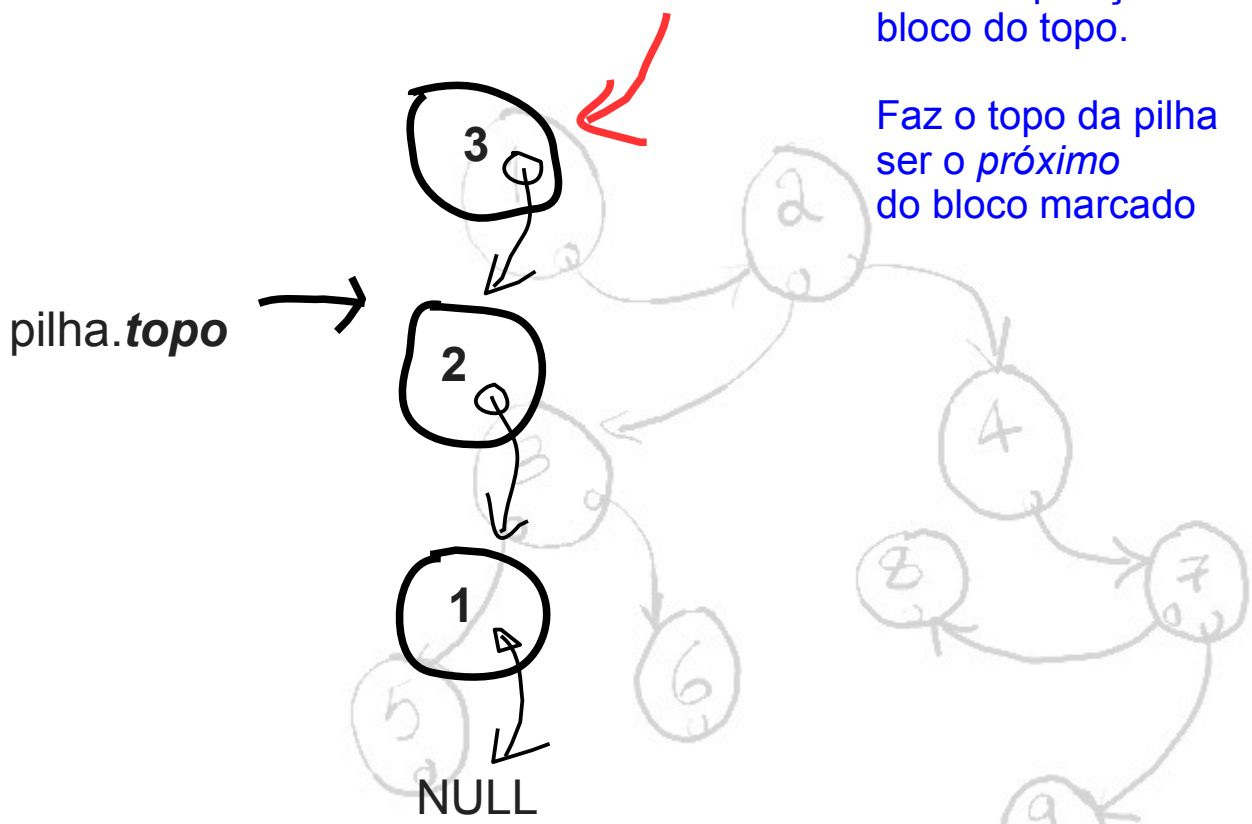
Faz o ponteiro do bloco apontar para o topo da pilha

Faz o topo apontar para o novo bloco

Dinâmica da Pilha

Elementos removidos:

POP(pilha):



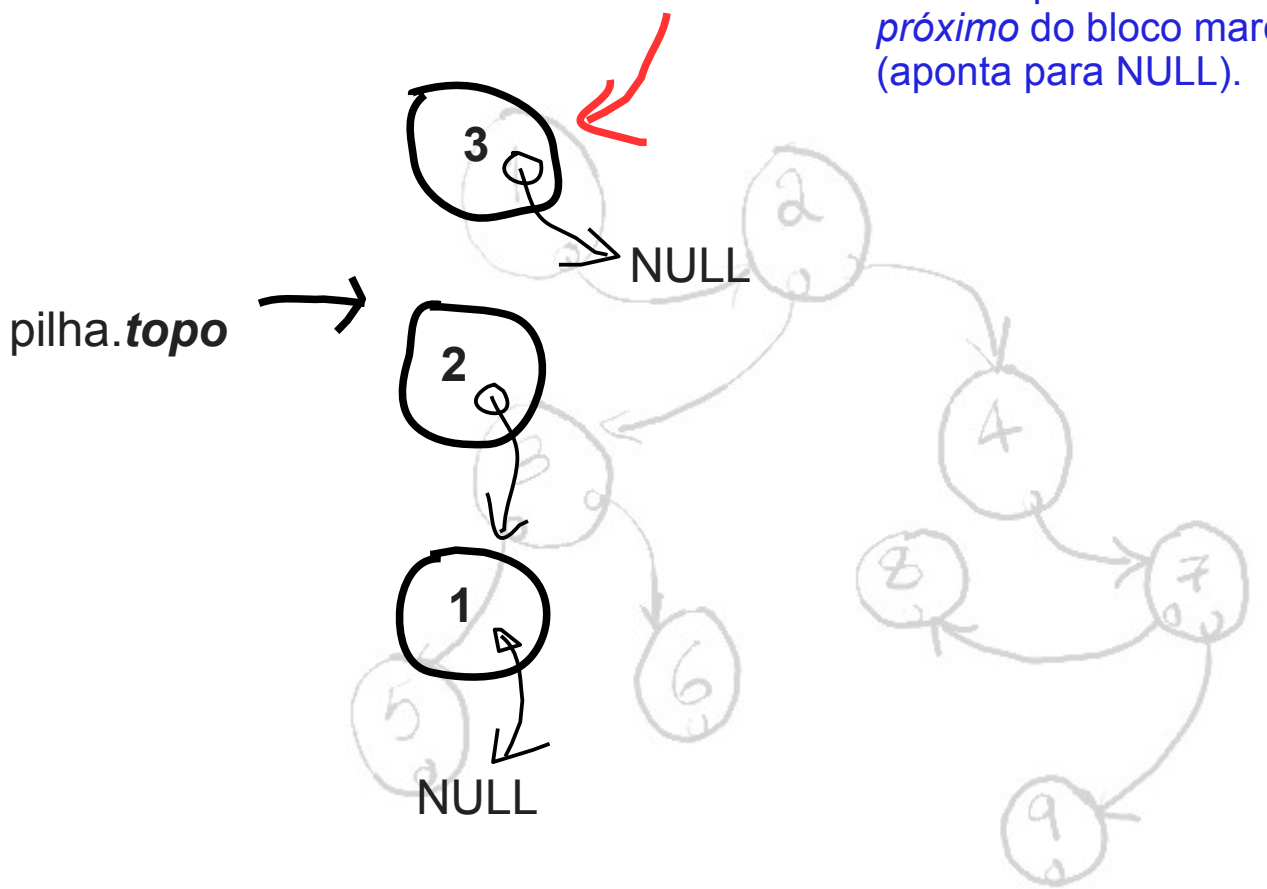
Marca a posição do bloco do topo.

Faz o topo da pilha ser o *próximo* do bloco marcado

Dinâmica da Pilha

Elementos removidos:

POP(pilha):

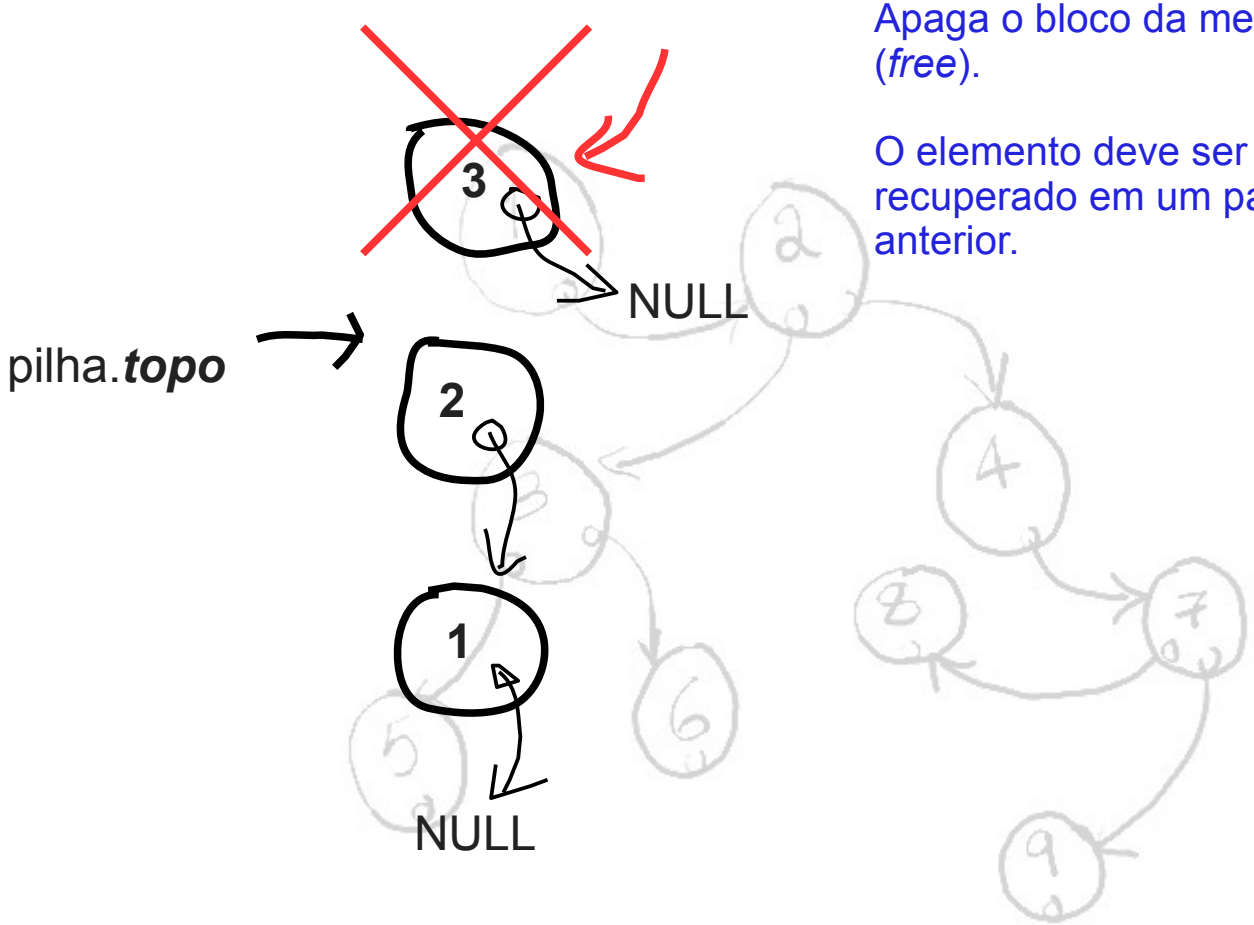


Libera o ponteiro de *próximo* do bloco marcado (aponta para NULL).

Dinâmica da Pilha

Elementos removidos: 3

POP(pilha):



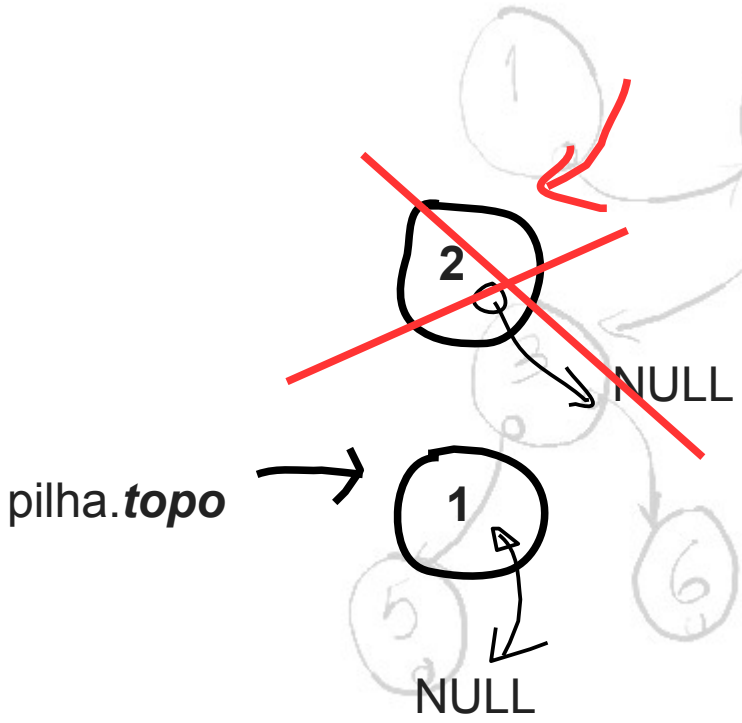
Apaga o bloco da memória (*free*).

O elemento deve ser recuperado em um passo anterior.

Dinâmica da Pilha

Elementos removidos: 3, 2

POP(pilha):



Marca o bloco.

Move o topo.

Faz o próximo apontar para NULL.

Apaga o bloco da memória (*free*).

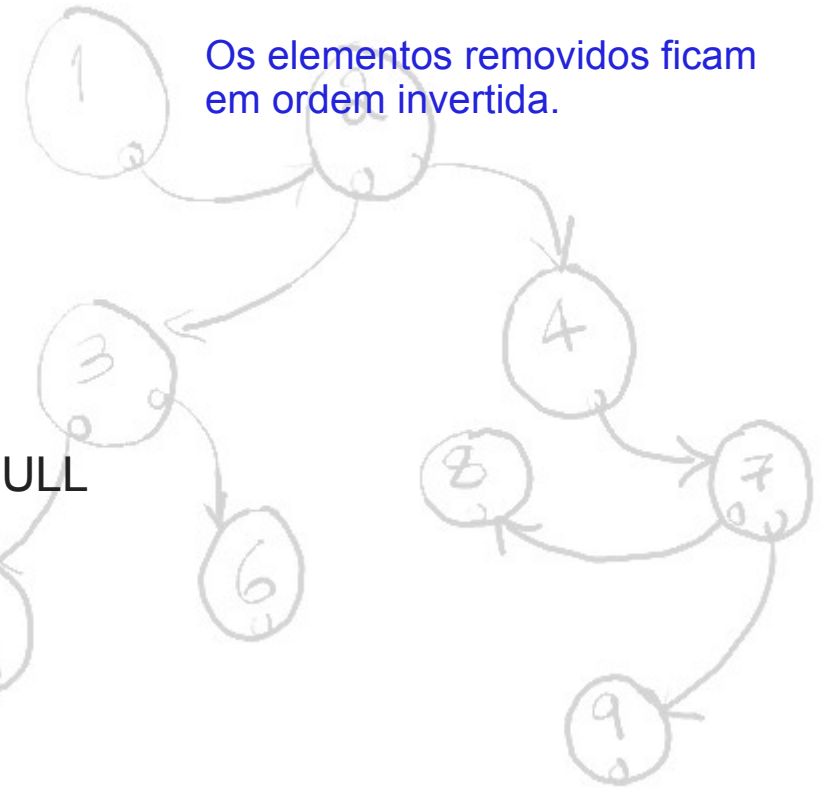
Dinâmica da Pilha

Elementos removidos: 3, 2, 1

POP(pilha):

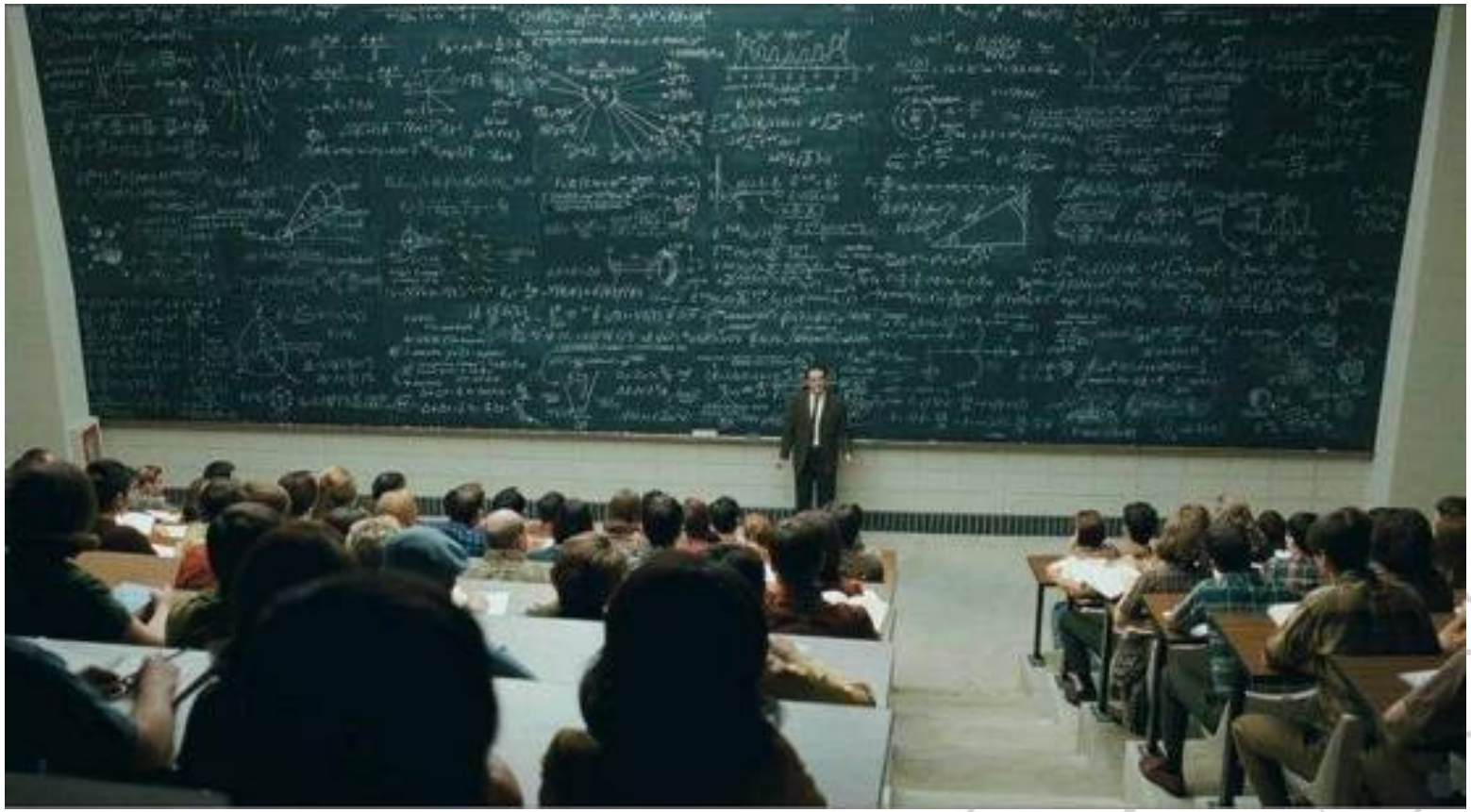
Ao remover o último elemento a pilha deve retornar ao seu estado inicial.

Os elementos removidos ficam em ordem invertida.



pilha.***topo*** → NULL

Implementação de Pilha



… para o quadro!