



*UNIVERSIDADE FEDERAL DE ITAJUBÁ*

# **Algoritmos e Estrutura de Dados I**

**COM 111**

**Pilha**

**Vanessa Cristina Oliveira de Souza**



# Introdução

---

- ▶ Em Ciência da Computação é fundamental o trabalho com conjuntos de dados que podem representar coleções de números, dados de um funcionário, dados de um produto, entre outros.
- ▶ Esses conjuntos são aqui chamados de **conjuntos dinâmicos**, pois os algoritmos que os manipulam fazem com que eles cresçam, encolham ou sofram alterações ao longo do tempo.





# Introdução

---

- ▶ Programas operam sobre dados
- ▶ Dados são **relacionados** e possuem **estrutura**





# Introdução

---

- ▶ O QUE VOCÊ PRECISA GUARDAR?
  - ▶ Dado
- ▶ COMO VOCÊ PRECISA GUARDAR?
  - ▶ Estrutura de Dados
- ▶ O QUE VOCÊ PRECISA FAZER COM ESSES DADOS?
  - ▶ Operações





# Introdução

---

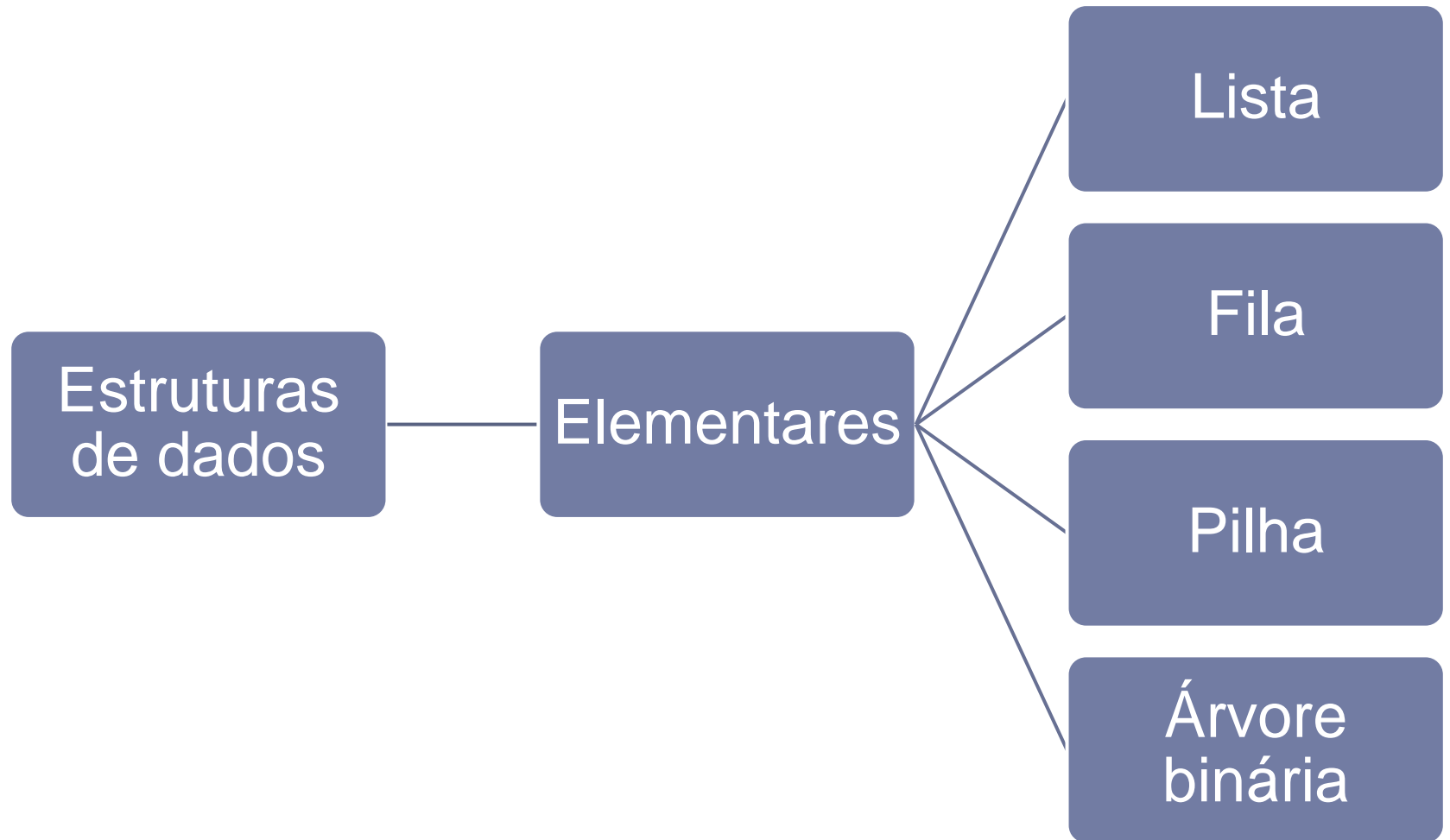
- ▶ Existem algumas Estruturas de Dados básicas importantes para a construção de algoritmos mais bem elaborados.





# Introdução

---





# Introdução

---

**QUAL ESTRUTURA ESCOLHER????**





# Introdução

---

- ▶ As principais operações sobre os conjuntos dinâmicos são:
  - ▶ Inserir um elemento
  - ▶ Excluir um elemento
  - ▶ Buscar um elemento
  - ▶ Encontrar o maior
  - ▶ Encontrar o menor
  - ▶ Contar os elementos da estrutura
  - ▶ Alterar os elementos da estrutura
  - ▶ Buscar o elemento sucessor e o predecessor







# Introdução

---

QUAL ESTRUTURA ESCOLHER????

***A forma como cada estrutura/TAD realiza as operações é que leva o desenvolvedor a escolher uma ou outra estrutura***





# Introdução

---

## ► Lista:

- Uma estrutura de dados do tipo lista é uma sequência finita de elementos ligados entre si

Lista Vazia





# Introdução

---

## ► Lista:

- Uma estrutura de dados do tipo lista é uma sequência finita de elementos ligados entre si

Insere Ana na lista

ANA





# Introdução

---

## ► Lista:

- Uma estrutura de dados do tipo lista é uma sequência finita de elementos ligados entre si

Insere João na lista

ANA

JOÃO





# Introdução

---

## ► Lista:

- Uma estrutura de dados do tipo lista é uma sequência finita de elementos ligados entre si



Insere Camila na lista



# Introdução

---

## ► Lista:

- Uma estrutura de dados do tipo lista é uma sequência finita de elementos ligados entre si



Insere Camila na lista



# Introdução

---

## ► Lista:

- Uma estrutura de dados do tipo lista é uma sequência finita de elementos ligados entre si

Insere Camila na lista

ANA

CAMILA

JOÃO



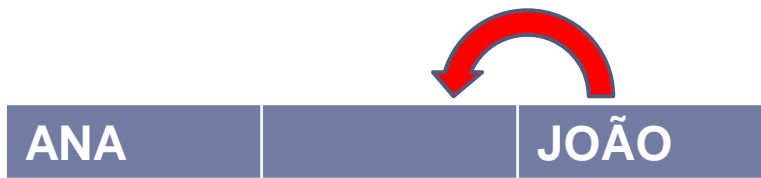


# Introdução

---

## ► Lista:

- Uma estrutura de dados do tipo lista é uma sequência finita de elementos ligados entre si



Remove Camila na lista





# Introdução

---

## ► Lista:

- Uma estrutura de dados do tipo lista é uma sequência finita de elementos ligados entre si

Remove Camila na lista

ANA

JOÃO





# Introdução

---

## ▶ Lista:

- ▶ Uma estrutura de dados do tipo lista é uma sequência finita de elementos ligados entre si
- ▶ Uma lista representa um conjunto de dados organizado em ordem linear.
- ▶ Em uma lista, não há regras de inserção, remoção e alteração dos elementos.
- ▶ São estruturas muito flexíveis e que possui diversas variantes, tais como:
  - ▶ Lista simplesmente encadeada (ordenada e não ordenada)
  - ▶ Lista duplamente encadeada (ordenada e não ordenada)
  - ▶ Lista circular





# Introdução

---

## ► Pilha e Fila:

- Pilha e Fila são tipos especiais de lista linear
- Pilha e Fila são, portanto, estruturas lineares
- Nelas as regras de manipulação dos elementos (inserção, remoção e alteração) são rígidas.
- Tal rigidez tornam as estruturas mais fáceis de serem implementadas



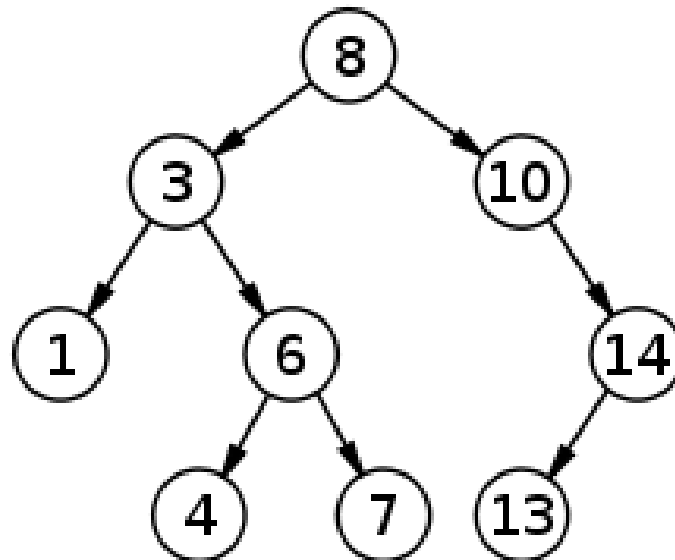


# Introdução

---

## ▶ Árvore Binária de Consulta:

- ▶ As estruturas do tipo árvore não são lineares, ou seja, os elementos que as compõem não estão armazenados de forma sequencial e também não estão encadeados.





# Introdução

---

- ▶ Nas próximas aulas, aprenderemos o que são pilhas e filas, bem como aprenderemos utilizá-las para resolver alguns problemas computacionais.





# Definições Importantes

---

- ▶ Listas, filas e pilhas podem ser:
  - ▶ Estáticas ou Dinâmicas
  - ▶ Homogêneas ou Heterogêneas



# Tipo Abstrato de Dados do tipo PILHA



- ▶ A estrutura denominada pilha é considerada uma lista do tipo LIFO : *Last In First Out*
  - ▶ *FILO : First In Last Out*
- ▶ Ou seja, o elemento inserido por último é o primeiro a ser removido da estrutura
  - ▶ o primeiro elemento inserido será o último a ser removido
- ▶ Também conhecida como *stack*







- ▶ Aplicações típicas do TAD Pilha
  - ▶ **Aplicações em estruturas aninhadas**
  - ▶ Uso por compiladores
    - ▶ Chamada de funções
    - ▶ Recursão
  - ▶ Análise de expressões e sintaxe
  - ▶ Mecanismo de desfazer/refazer dos editores de texto
  - ▶ Navegação entre páginas Web
- ▶ Calculadores que utilizam a Notação Polonesa Inversa
  - ▶  $((1 + 2) * 4) + 3$  em notação pós-fixa  $\rightarrow 1\ 2\ +\ 4\ *\ 3\ +$





## ▶ Estrutura de Dados

- ▶ LIFO
- ▶ O último elemento a ser inserido será o primeiro elemento a ser retirado

## ▶ Operações

- ▶ CriaPilha (cria a estrutura pilha vazia)
- ▶ Push (inserir elemento na pilha)
- ▶ Pop (remover elemento da pilha)
- ▶ Top (mostrar quem está no topo da pilha)
- ▶ Esvazia (remove todos os elementos da pilha)
- ▶ pilhaVazia (verifica se a pilha está vazia)
- ▶ pilhaCheia (verifica se a pilha está cheia – estruturas estáticas)

## ▶ Implementação

- ▶ Vetor
  - ▶ Lista
- 



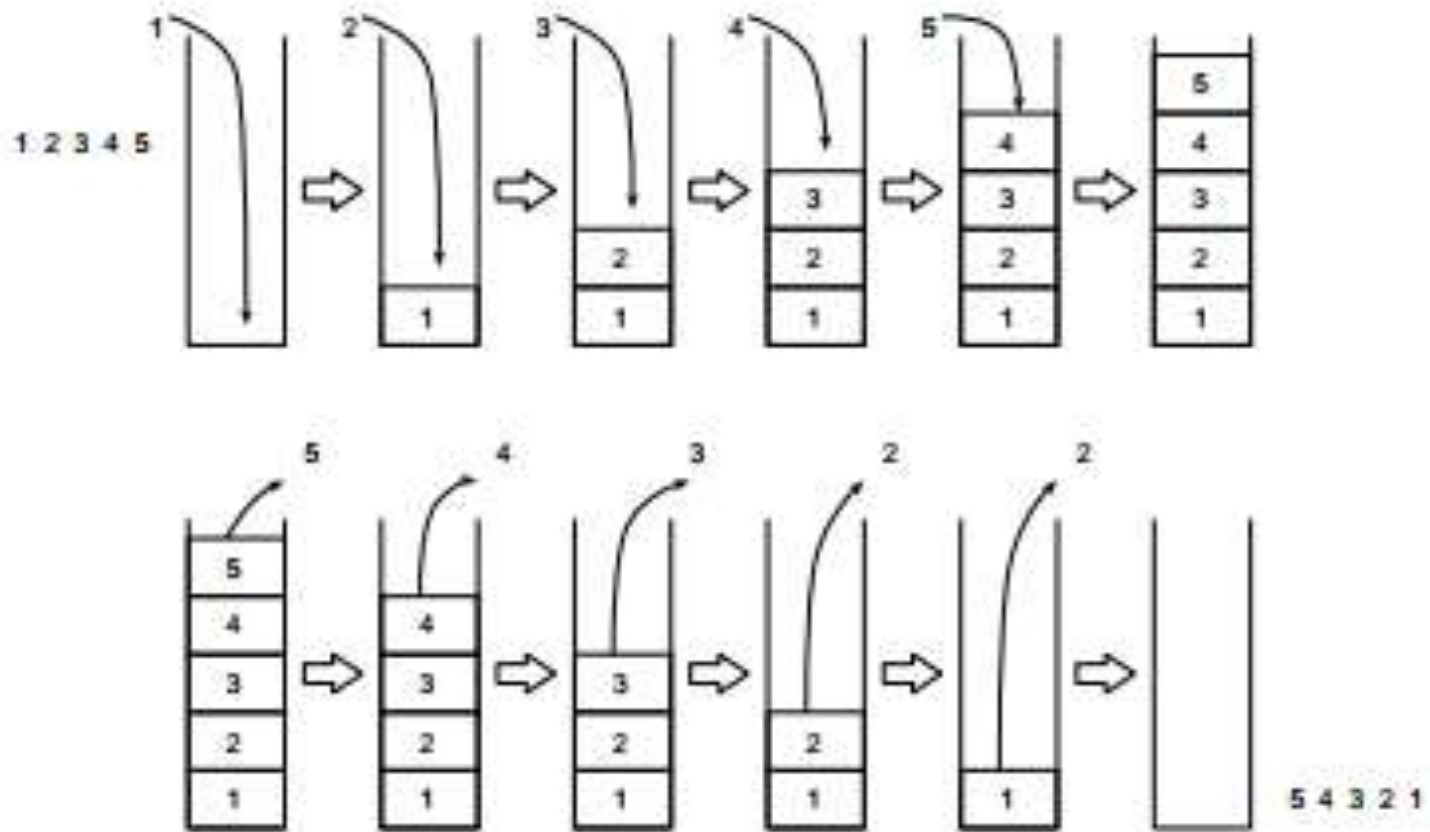


- ▶ Qualquer estrutura desse tipo possui um ponteiro denominado TOPO, na qual todas as operações de inserção e remoção acontecem.
- ▶ As operações acontecem sempre na mesma extremidade da estrutura.



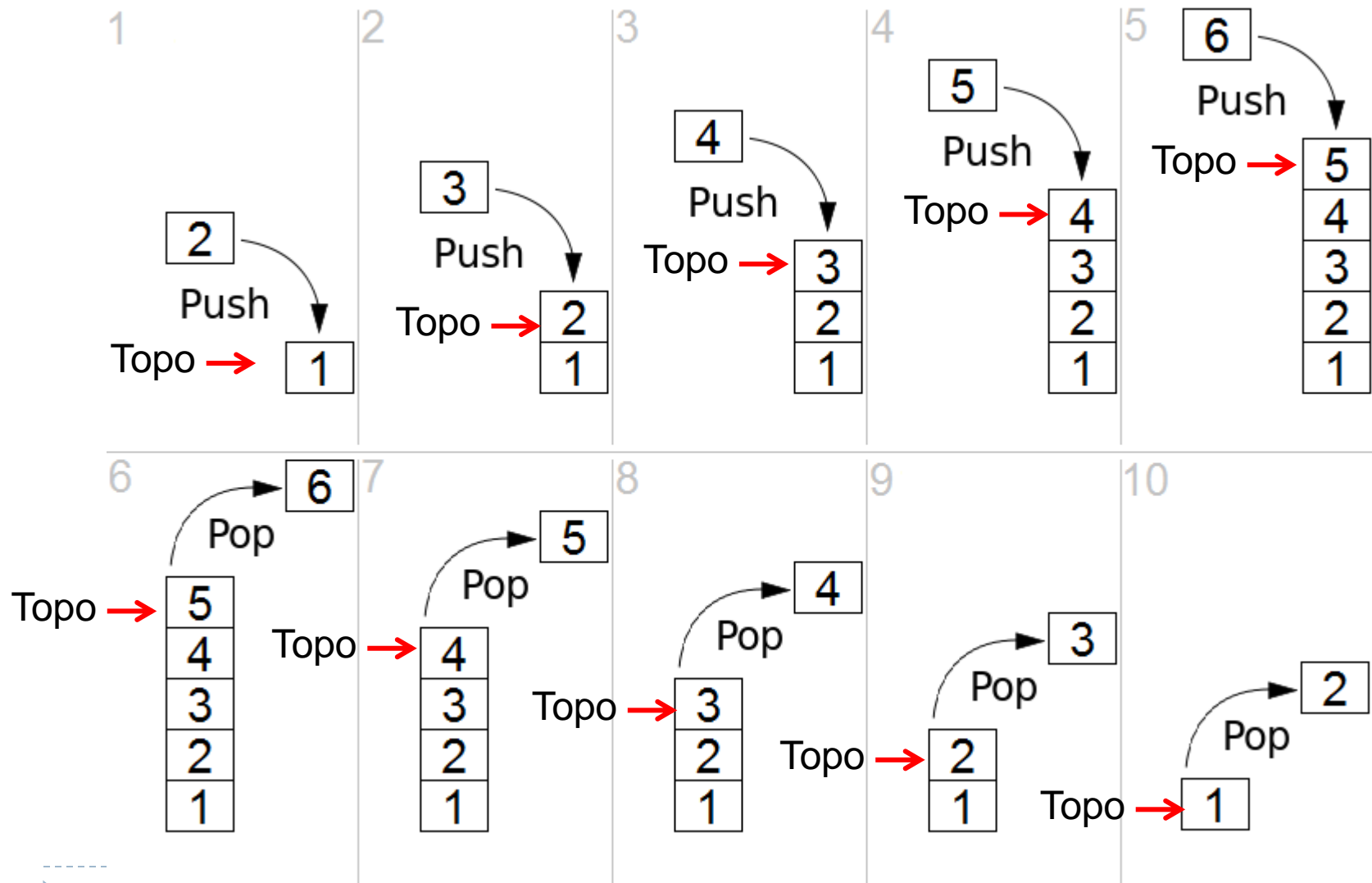


# Pilha





# Pilha





- ▶ Existem várias opções de estruturas de dados que podem representar pilhas.
- ▶ As duas representações mais utilizadas são as implementações por meio de arranjos e de estruturas auto-referenciadas.
  - ▶ Arranjos -> Pilhas Estáticas
  - ▶ Estruturas auto-referenciadas -> Pilhas Dinâmicas



A vertical decorative bar in a light blue color, positioned on the left side of the slide.

# PILHA ESTÁTICA



## Pilhas Estáticas

---

- ▶ Em uma implementação por meio de arranjos (vetores) os itens da pilha são armazenados em posições contíguas de memória.
  - ▶ Os itens são armazenados em um arranjo de tamanho suficiente para conter a pilha
  - ▶ Há uma referência para o item no topo da pilha
  - ▶ Há uma constante para guardar o tamanho máximo permitido na pilha

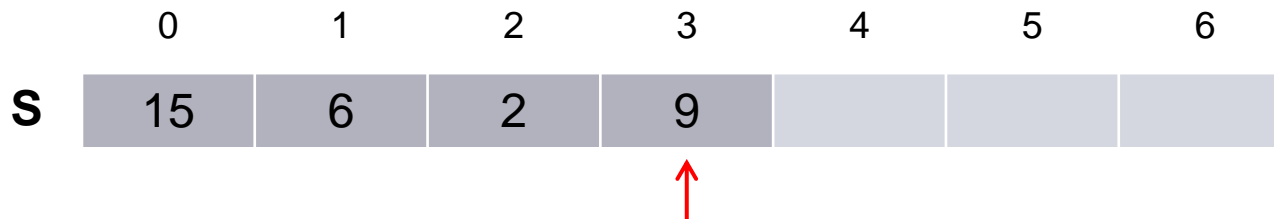






# Pilhas Estáticas

---



$\text{topo}(S) = 3$





# Pilhas Estáticas

---

push(S, 17)

	0	1	2	3	4	5	6
<b>S</b>	15	6	2	9	17		



topo(S) = 4





# Pilhas Estáticas

---

push(S, 3)

	0	1	2	3	4	5	6
<b>S</b>	15	6	2	9	17	3	



topo(S) = 5





# Pilhas Estáticas

pop(S)

	0	1	2	3	4	5	6
<b>S</b>	15	6	2	9	17	3	

↑  
 $\text{topo}(S) = 4$

- A pilha S, após a chamada POP(S) retornou o elemento 3, que é o elemento mais recente inserido na pilha.
- O topo é atualizado e o elemento do topo é o 17.
- Embora o elemento 3 ainda apareça no arranjo, ele não está mais na pilha.



# Pilhas Estáticas

---

- ▶ Considerando uma pilha de inteiros, e as funções abaixo listadas, definiremos a interface do TAD Pilha
  - ▶ CriaPilha (cria a estrutura pilha vazia)
  - ▶ Push (inserir elemento na pilha)
  - ▶ Pop (remover elemento da pilha)
  - ▶ Topo (mostrar quem está no topo da pilha)
  - ▶ Esvazia (remove todos os elementos da pilha)
  - ▶ pilhaVazia (verifica se a pilha está vazia)
  - ▶ pilhaCheia (verifica se a pilha está cheia – estruturas estáticas)

```
struct pilha
{
    int arranjo[tamMax];
    int topo;
};
```



# Pilhas Estáticas

---

- ▶ Considerando o TAD Pilha Estática definido, vamos implementar as operações
  - ▶ CriaPilha (cria a estrutura pilha vazia)
  - ▶ Push (inserir elemento na pilha)
  - ▶ Pop (remover elemento da pilha)
  - ▶ Topo (mostrar quem está no topo da pilha)
  - ▶ Esvazia (remove todos os elementos da pilha)
  - ▶ pilhaVazia (verifica se a pilha está vazia)
  - ▶ pilhaCheia (verifica se a pilha está cheia – estruturas estáticas)

```
struct pilha
{
    int arranjo[tamMax];
    int topo;
};
```



# Pilhas Estáticas

---

- ▶ Considerando o TAD Pilha Estática definido, vamos implementar as operações
  - ▶ CriaPilha (cria a estrutura pilha vazia)
  - ▶ Push (inserir elemento na pilha)
  - ▶ Pop (remover elemento da pilha)
  - ▶ Topo (mostrar quem está no topo da pilha)
  - ▶ Esvazia (remove todos os elementos da pilha)
  - ▶ pilhaVazia (verifica se a pilha está vazia)
  - ▶ pilhaCheia (verifica se a pilha está cheia – estruturas estáticas)
  - ▶ **ListaDados (lista todos os elementos da pilha)**

```
struct pilha
{
    int arranjo[tamMax];
    int topo;
};
```

