



UNIVERSIDADE FEDERAL DE ITAJUBÁ

Algoritmos e Estrutura de Dados I

CTCO-01

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, ou dados do próprio sistema.
- ▶ 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





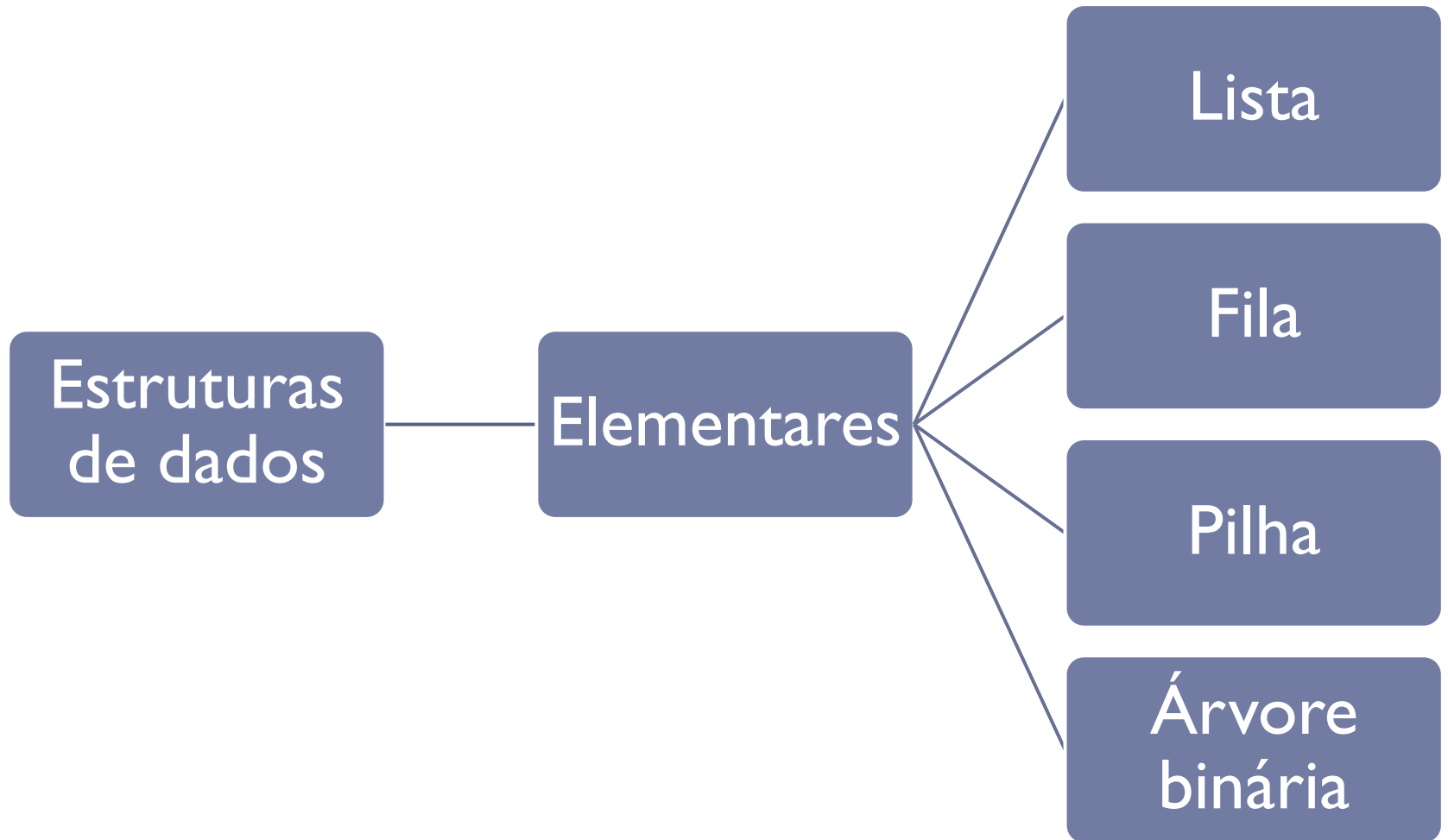
Introdução

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



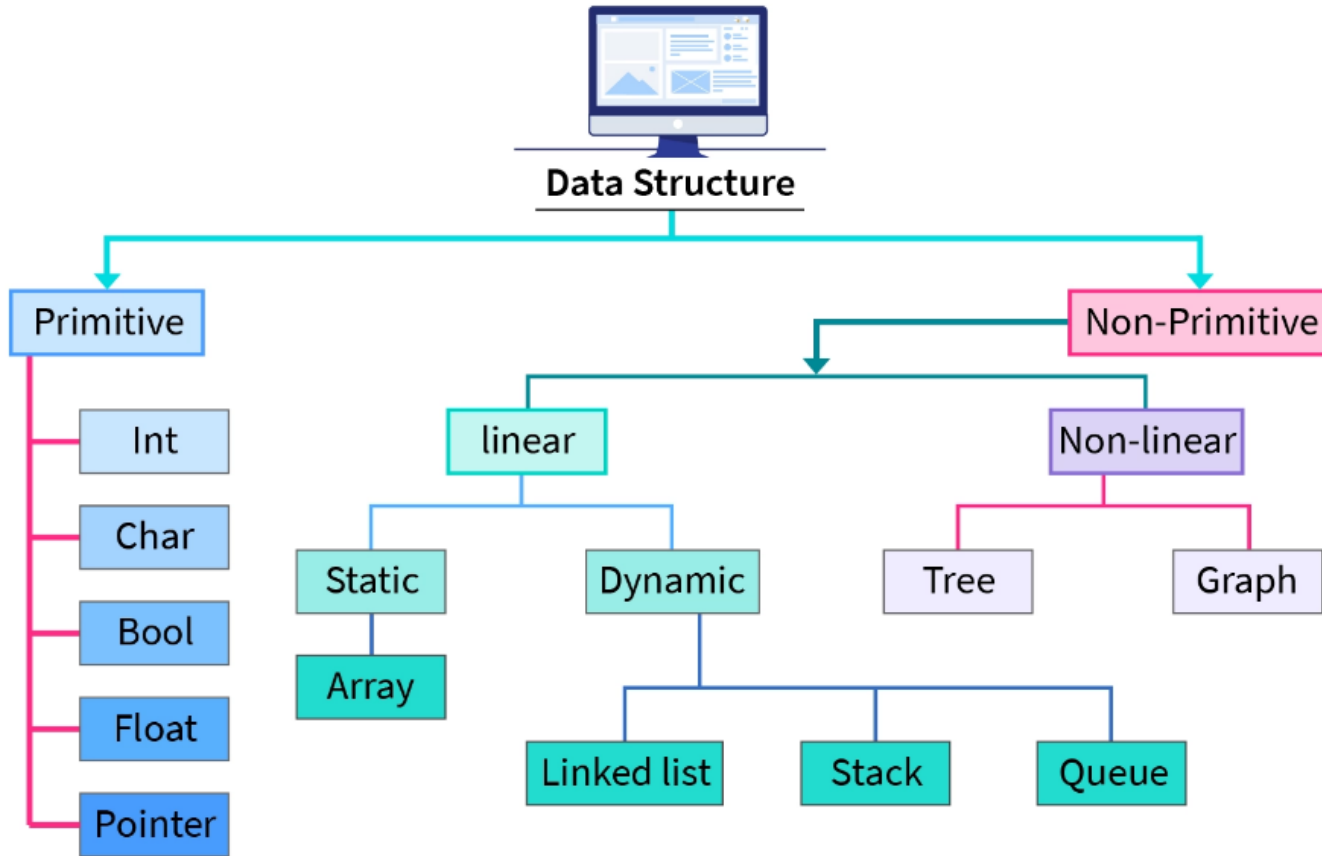


Introdução





Introdução



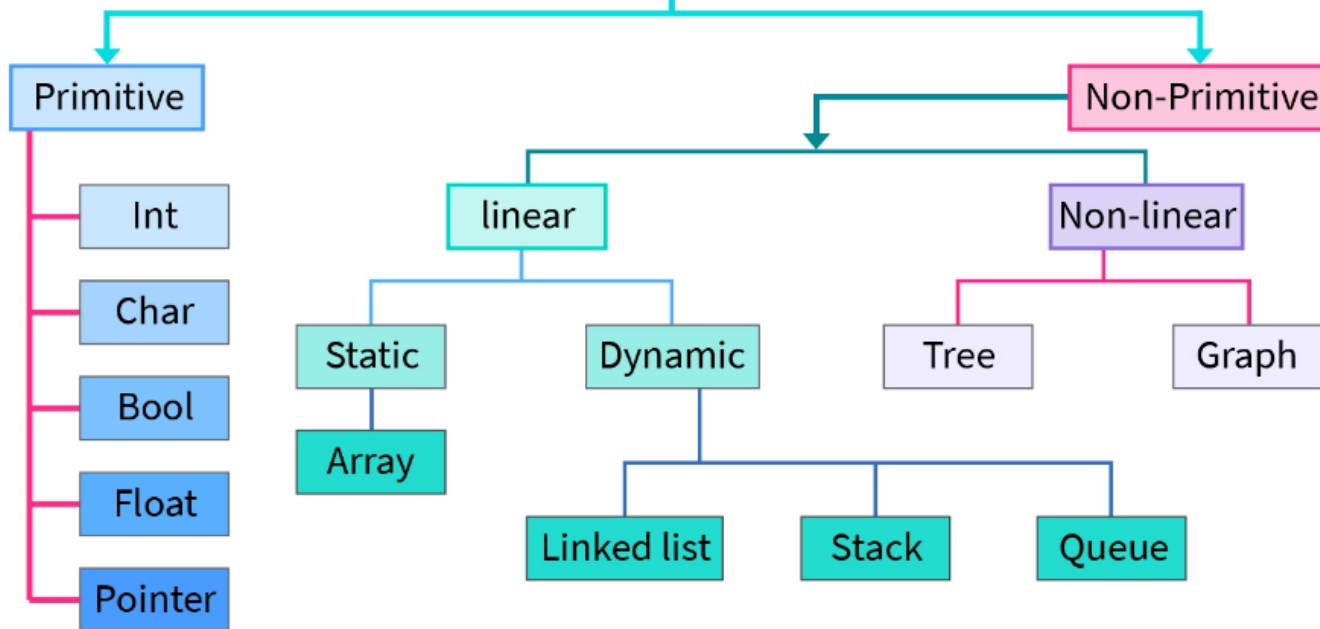


Introdução

Representam dados simples e fundamentais, como números ou caracteres, que servem de base para a construção de estruturas de dados mais complexas.

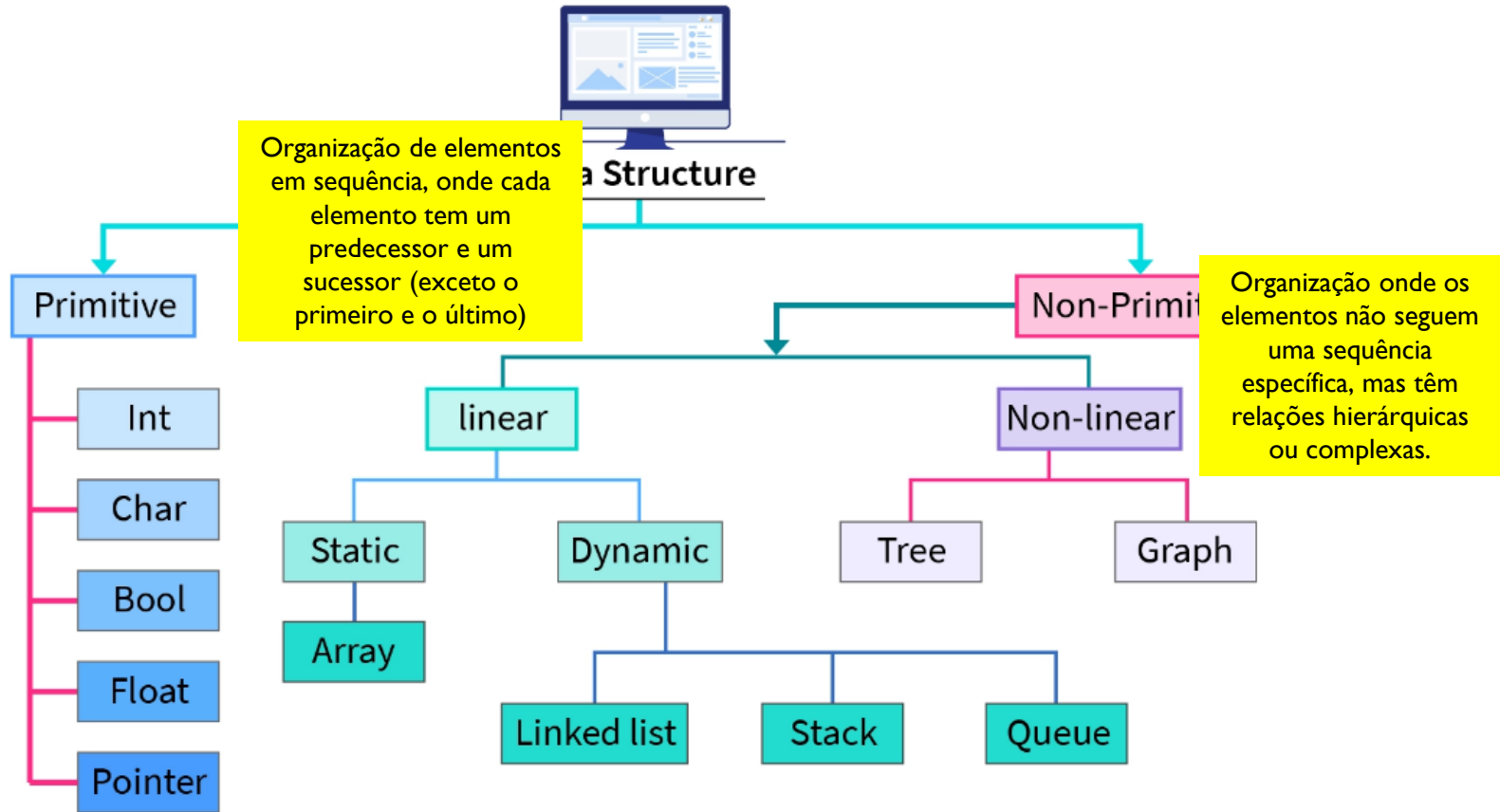


- ✓ Estruturas mais complexas, formadas a partir de estruturas primitivas ou outras estruturas não primitivas.
- ✓ Não são fornecidas diretamente pela linguagem de programação
- ✓ São criadas ou implementadas para organizar e manipular grandes volumes de dados de maneira eficiente.



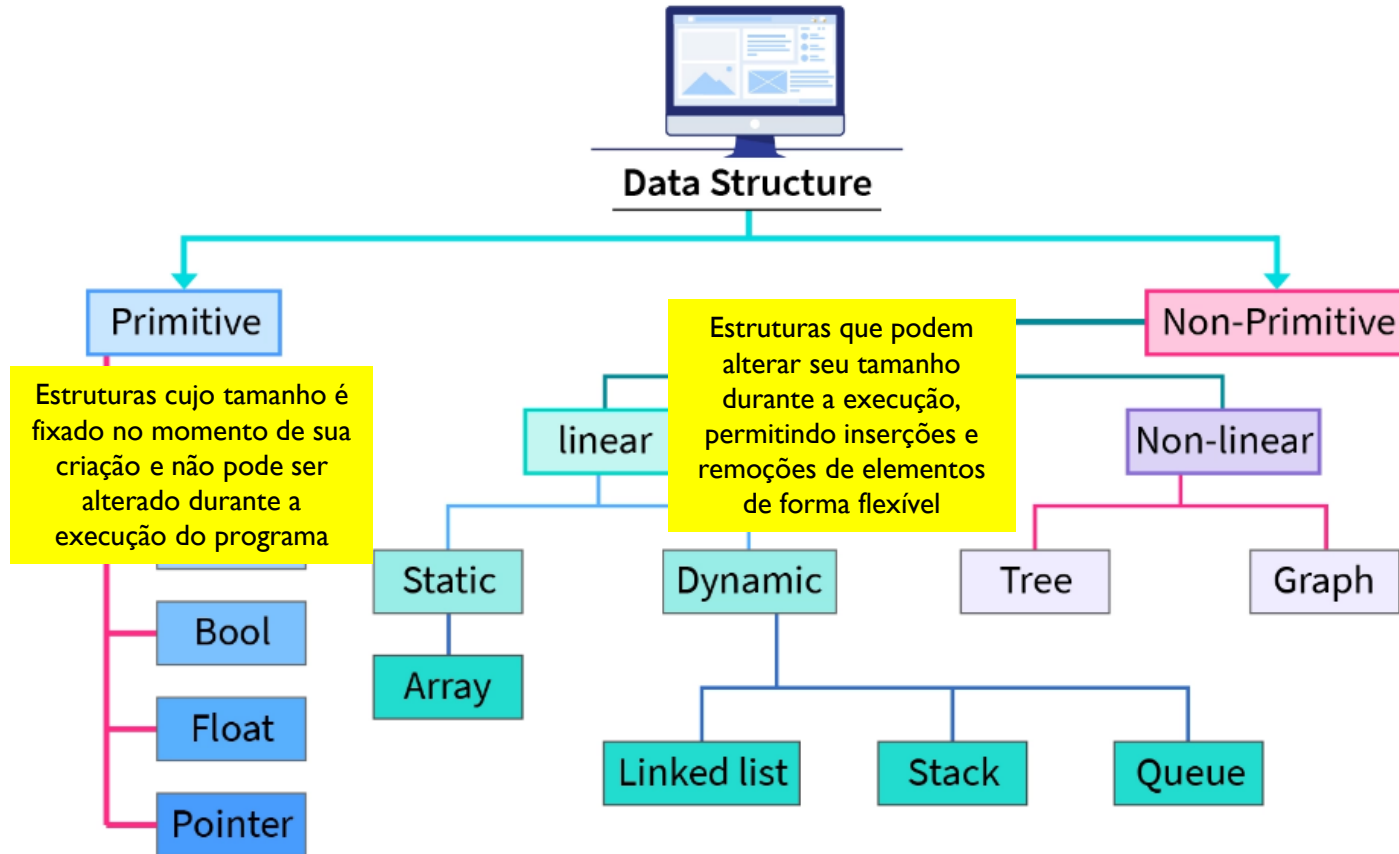


Introdução





Introdução





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





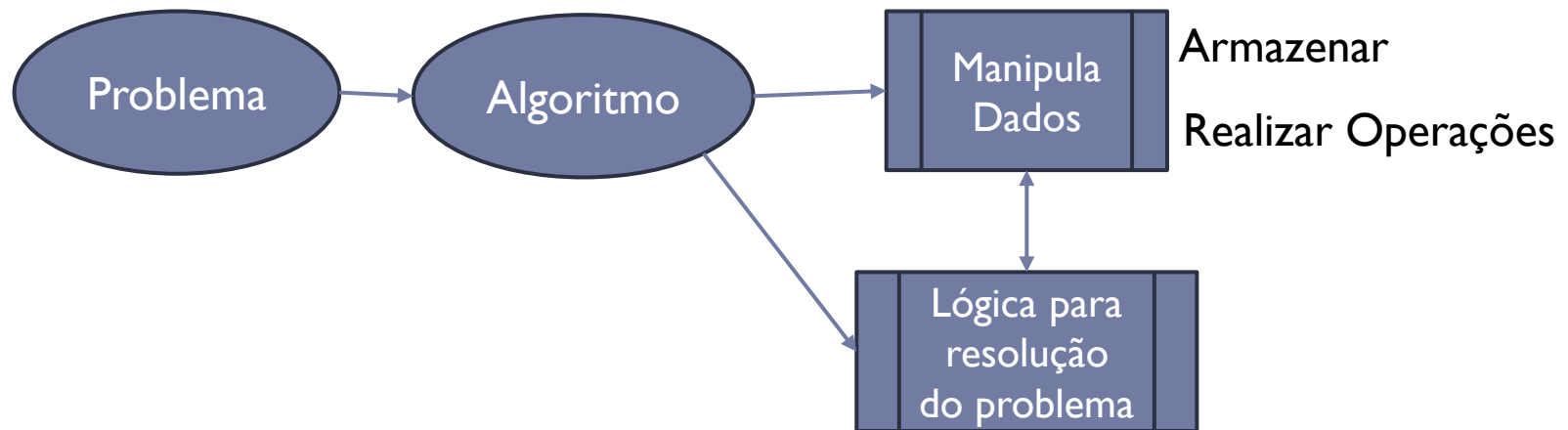
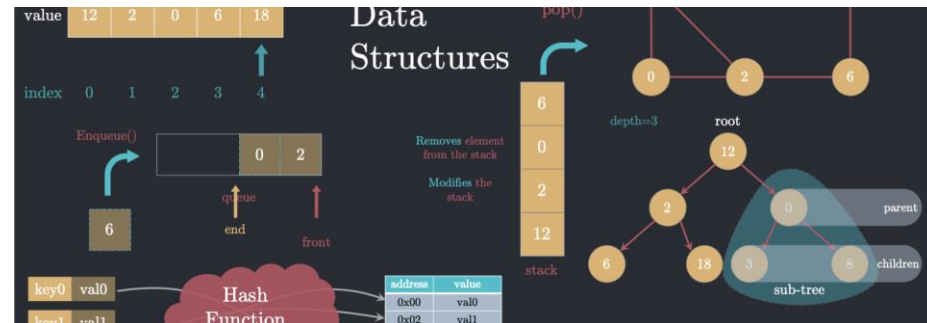
Introdução

QUAL ESTRUTURA ESCOLHER????



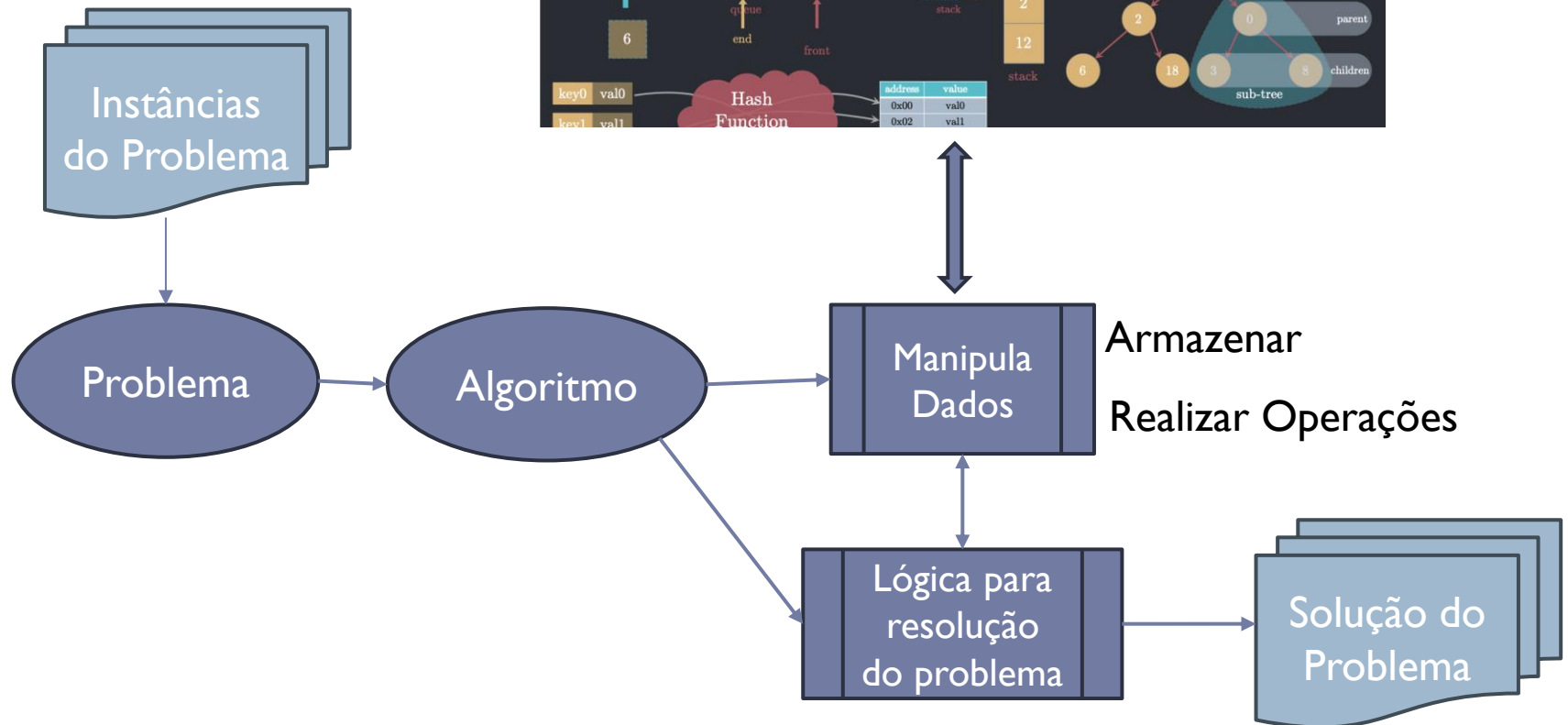
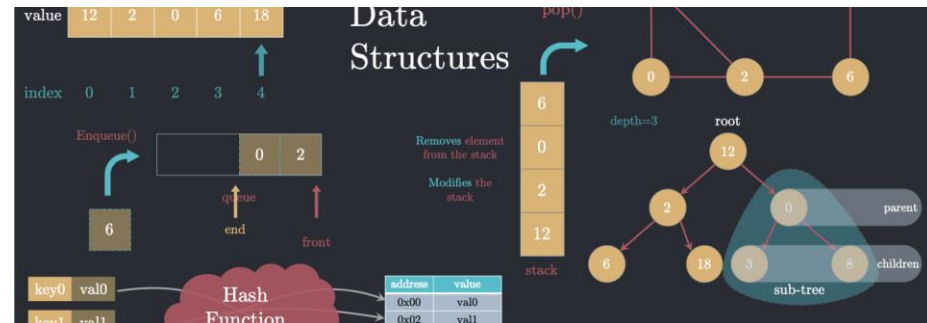


Introdução



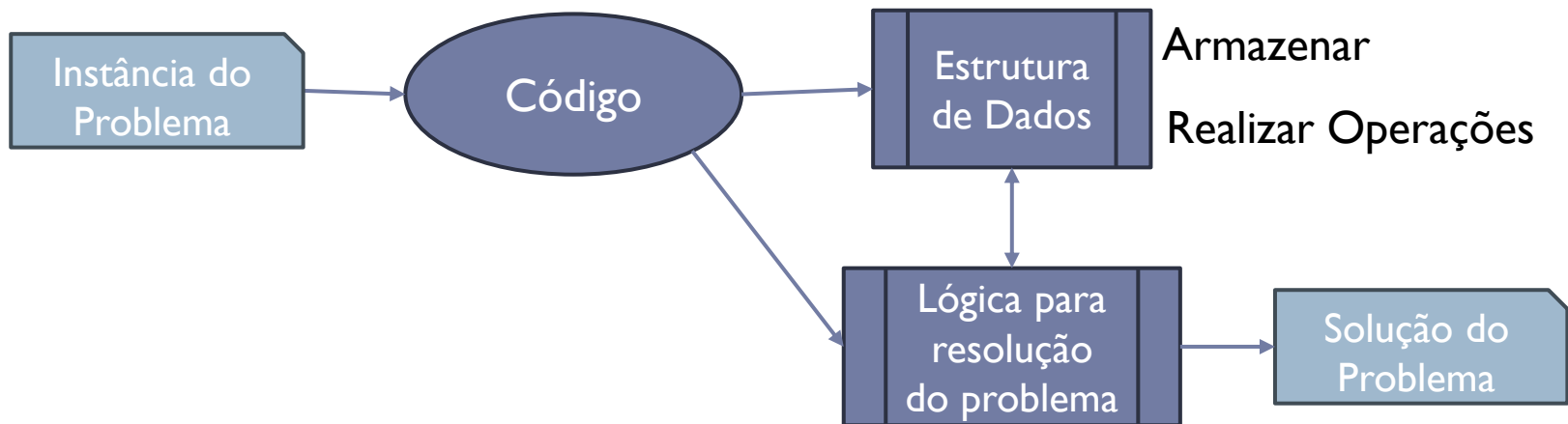


Introdução



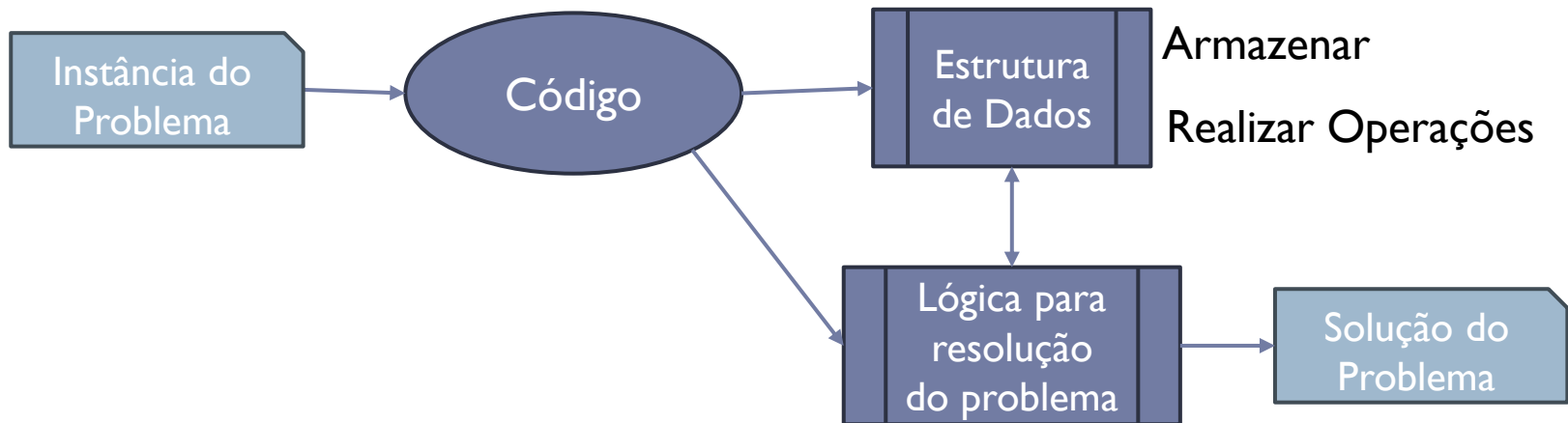
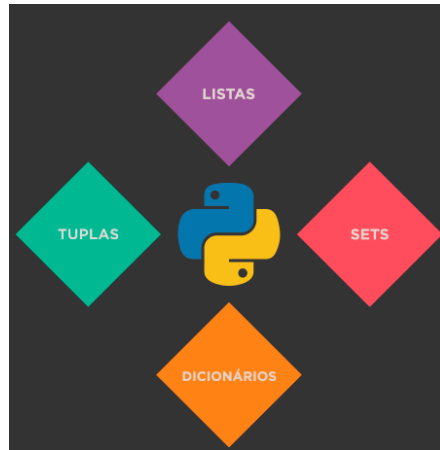


Introdução





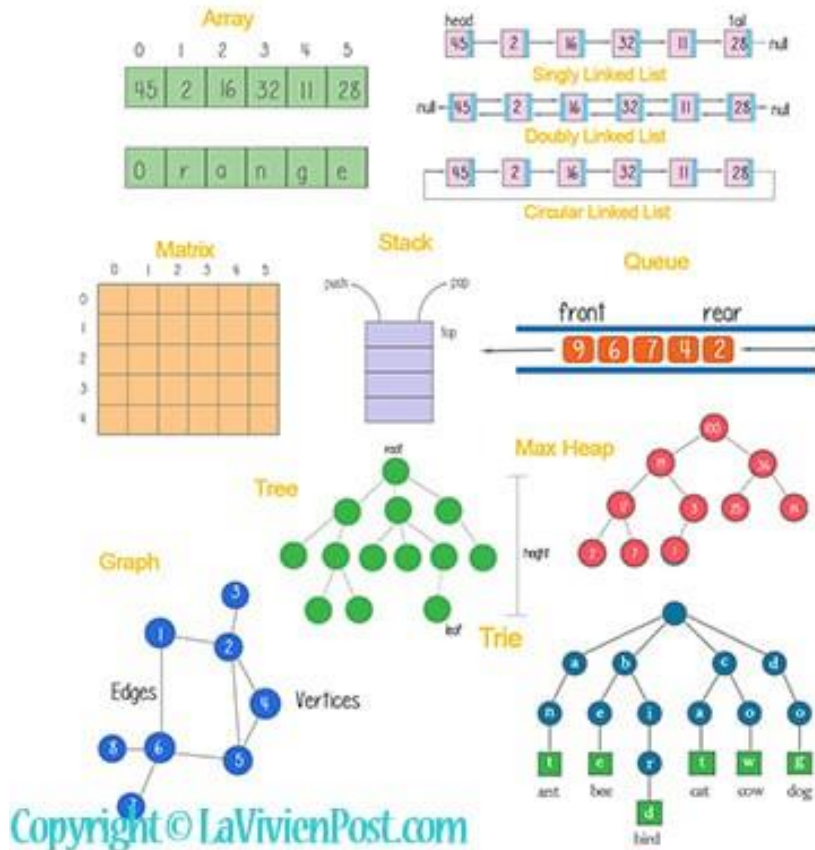
Introdução



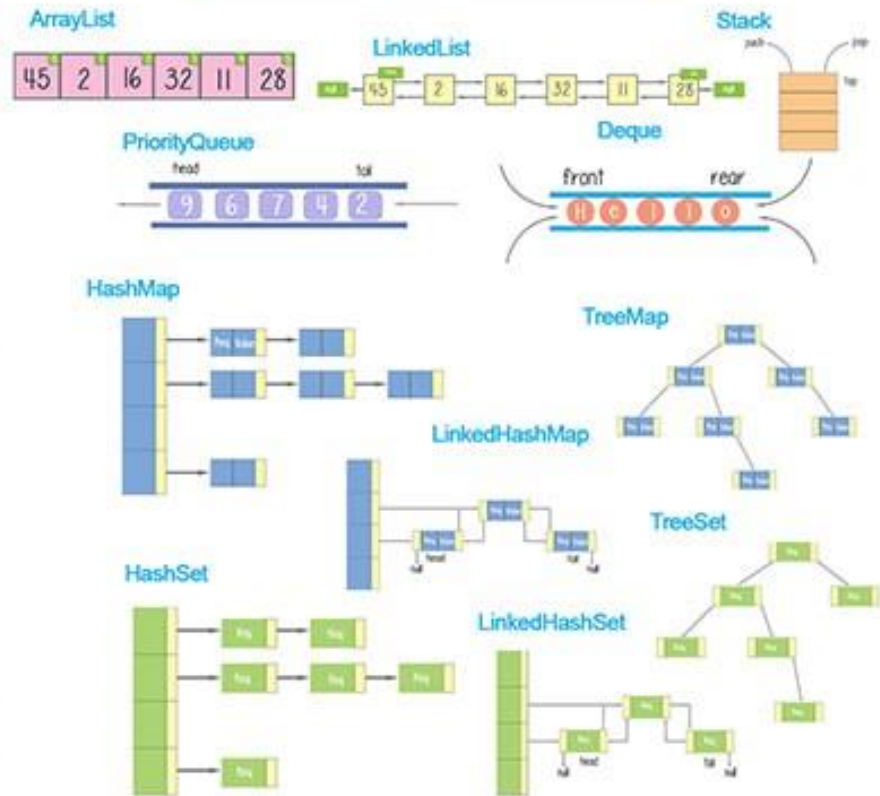


Introdução

Data structures



Java APIs



<https://www.linkedin.com/pulse/data-structures-java-omar-ismail/>



Introdução

<https://www.linkedin.com/pulse/data-structures-java-omar-ismail/>

Types of Data Structure in Java

There are some common types of data structure in Java they are as follows –

1. Arrays
2. Linked Lists
3. Stack
4. Queue
5. Graph
6. Set





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

Inserir Ana na lista

ANA





Introdução

► Lista:

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

Inserir João na lista

ANA

JOÃO





Introdução

► Lista:

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



Inserir Camila na lista



Introdução

► Lista:

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



Inserir Camila na lista



Introdução

► Lista:

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

Inserir Camila na lista

ANA

CAMILA

JOÃO

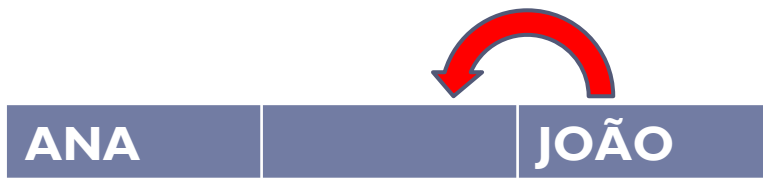




Introdução

► 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

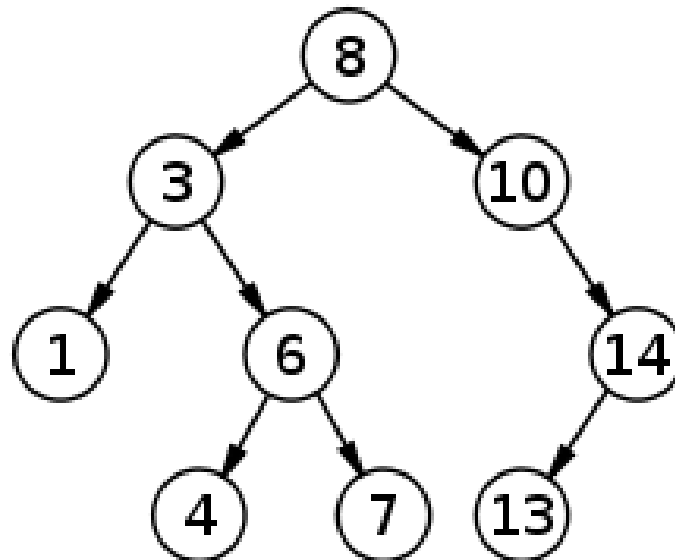




Introdução

▶ 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.





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*
 - ▶ *Stack Overflow*
 - ▶ *Full Stack*





- ▶ Aplicações típicas do TAD Pilha
 - ▶ Uso por compiladores
 - ▶ Chamada de funções
 - ▶ Recursão
 - ▶ Análise de expressões e sintaxe
 - ▶ balanceamento de parênteses
 - ▶ Mecanismo de desfazer/refazer dos editores de texto
 - ▶ Navegação entre páginas Web
 - ▶ Resolução de problemas de *backtracking*
 - ▶ Sudoku
 - ▶ busca em profundidade
 - ▶ resolução de labirintos
 - ▶ Calculadoras que utilizam a Notação Polonesa Inversa
 - ▶ $((1 + 2) * 4) + 3$ em notação pós-fixa -> $1\ 2\ +\ 4\ *\ 3\ +$





▶ 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)



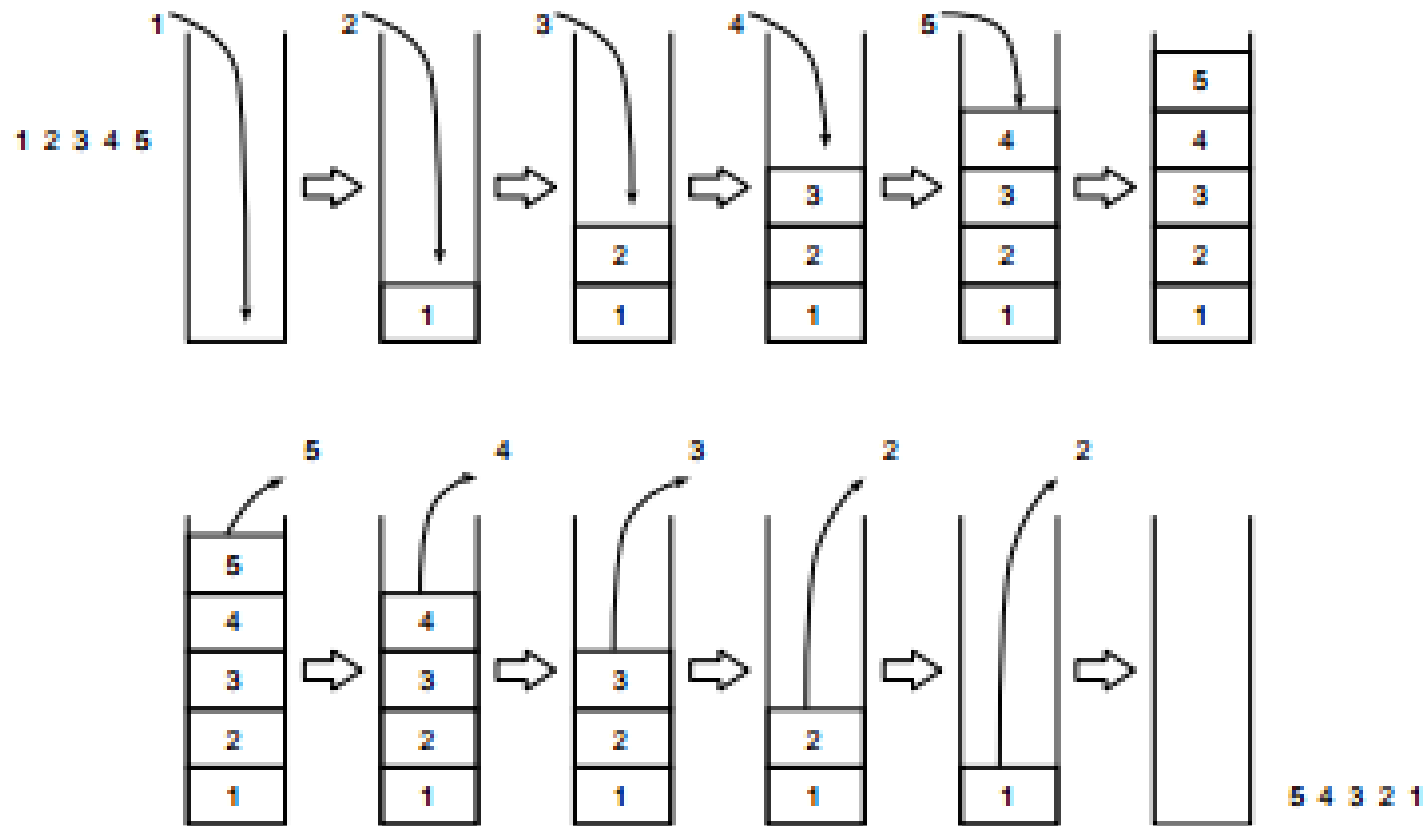


- ▶ Qualquer estrutura desse tipo possui um ponteiro denominado TOPO, na qual todas as operações de inserção e remoção acontecem.
- ▶ As operações acontecem sempre na mesma extremidade da estrutura.



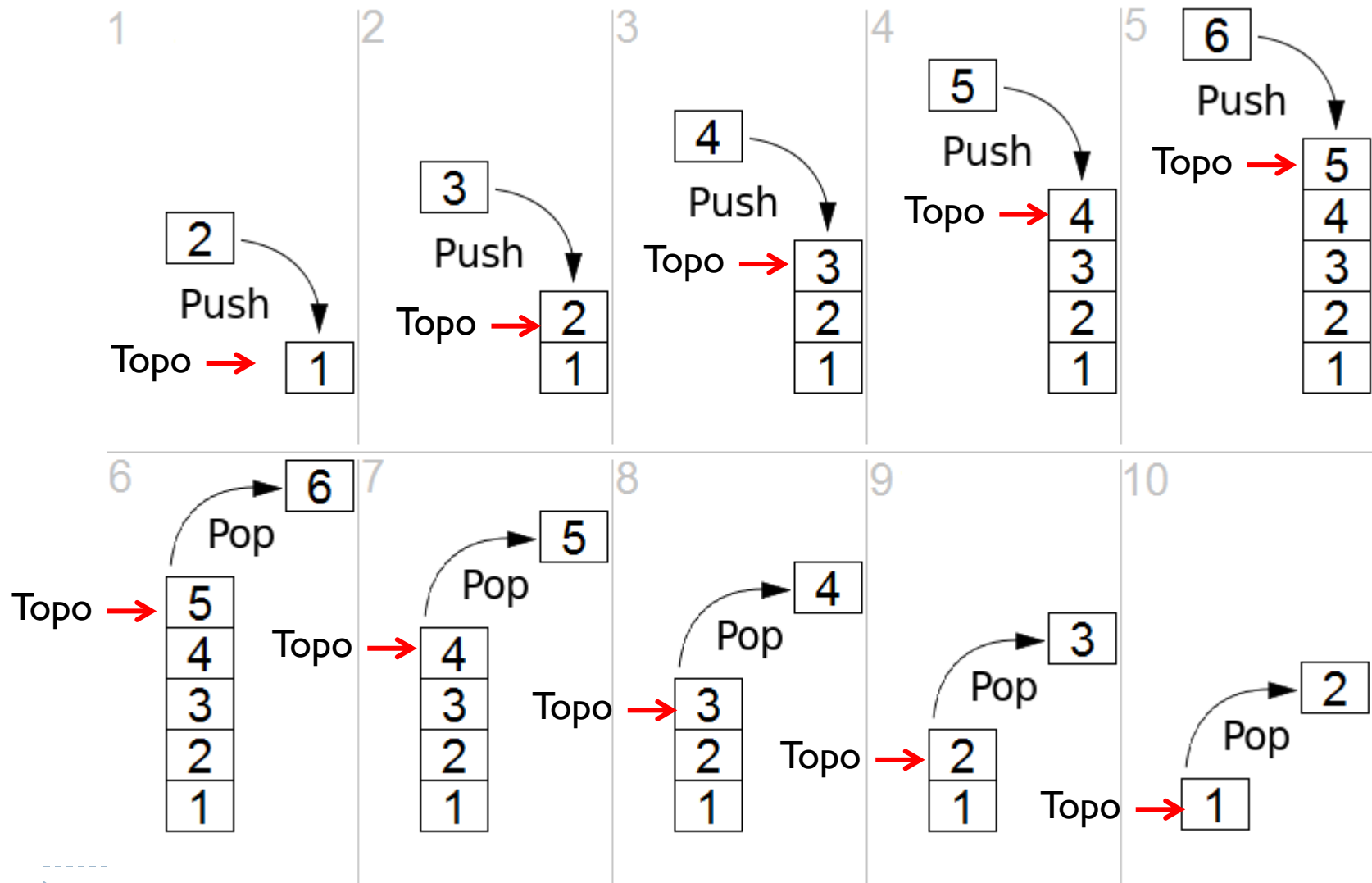


Pilha





Pilha



Pilha

- ▶ Pilha vazia
- ▶ Push
- ▶ Pop
- ▶ Top

- ▶ Busca por algum elemento na pilha
- ▶ Maior elemento
- ▶ Menor elemento





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





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 array 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

	0	1	2	3	4	5	6
S	15	6	2	9			



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





Pilhas Estáticas

push(S, 17)

	0	1	2	3	4	5	6
S	15	6	2	9	17		



$\text{topo}(S) = 4$





Pilhas Estáticas

push(S, 3)

	0	1	2	3	4	5	6
S	15	6	2	9	17	3	



topo(S) = 5





Pilhas Estáticas

pop(S)

	0	1	2	3	4	5	6
S	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 array, 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)
 - ▶ desalocaPilha (desaloca a estrutura)





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)
 - ▶ desalocaPilha (desaloca a estrutura)





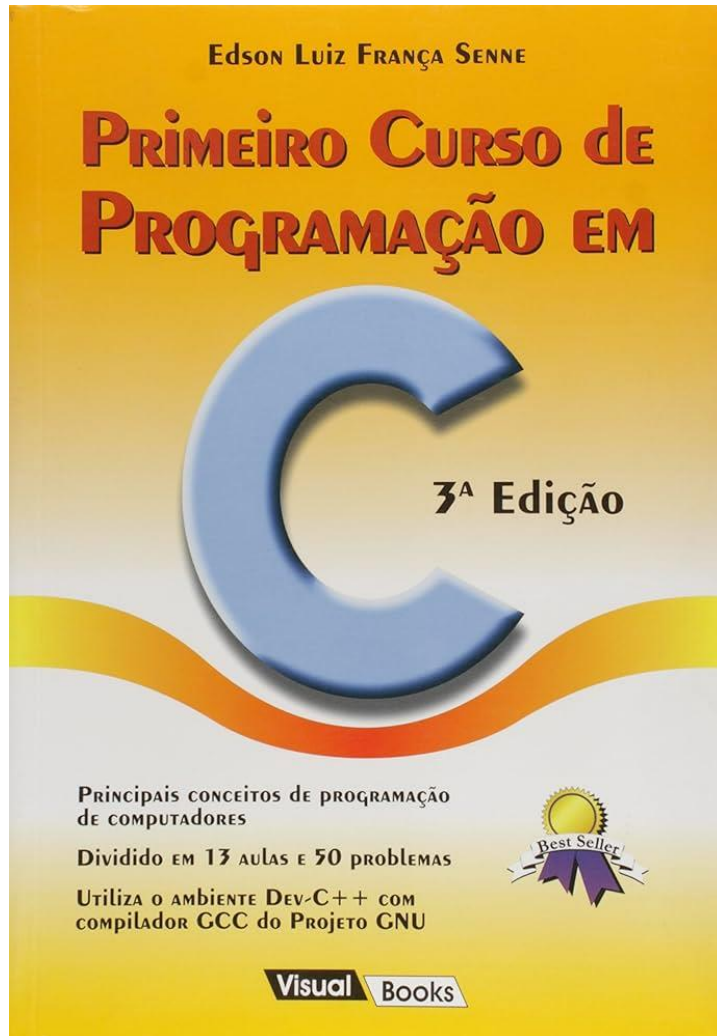
Pilhas Estáticas

- ▶ **Exercício: Verificação de sequência de números invertidos**
- ▶ Implemente um programa em C que leia uma sequência de números inteiros do usuário e armazene-os em uma pilha. Em seguida, o programa deve desempilhar os números e exibi-los na ordem inversa àquela em que foram inseridos.
- ▶ **Requisitos:**
 1. O programa deve solicitar que o usuário informe quantos números deseja inserir.
 2. Em seguida, o usuário deve fornecer os números inteiros um por um.
 3. Utilize uma pilha para armazenar os números.
 4. Após a inserção, o programa deve desempilhar e imprimir os números na ordem inversa à que foram inseridos.

- ▶ **Exemplo de execução:**

```
Digite o tamanho da pilha: 5
Digite o valor a ser inserido na pilha: 1
Digite o valor a ser inserido na pilha: 2
Digite o valor a ser inserido na pilha: 3
Digite o valor a ser inserido na pilha: 4
Digite o valor a ser inserido na pilha: 5
5 4 3 2 1
```

Leitura Recomendada



► Aula 13

- A codificação é dinâmica, mas a explicação é a mesma para estrutura estática
- Não se apegar à implementação