Estrutura de Dados (CC4652)

Aula 10 – Árvores Binárias de Busca

Prof. Luciano Rossi
Prof. Leonardo Anjoletto Ferreira
Prof. Flavio Tonidandel
Prof. Fabio Suim

Ciência da Computação
Centro Universitário FEI

2º Semestre de 2025

Árvores

Classificação das Estruturas de Dados

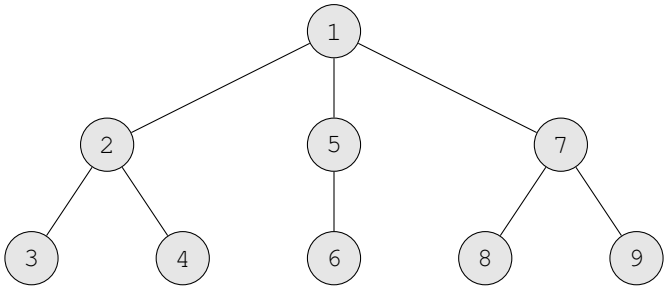
Primitivas	Compostas		
	Simples	Lineares	Não-Lineares
Inteiro	Cadeia (string)	Pilha	**Árvore**
Real	Registro (struct)	Fila	Grafo
Lógico	Arranjo (array)	Lista	
Caractere			

Árvores

centro universitário FEI

Árvores

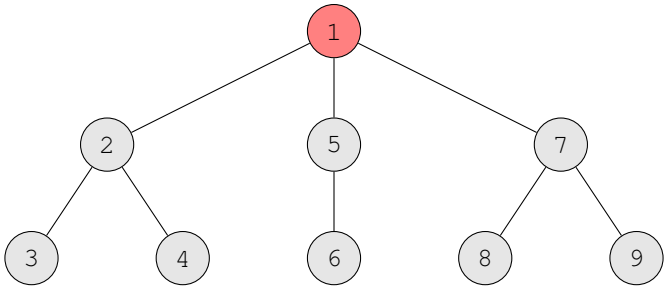
- Um **grafo** é uma estrutura matemática usada para modelar relações paritárias entre objetos.
- Uma **árvore** é um grafo acíclico conexo;



centro universitário FEI

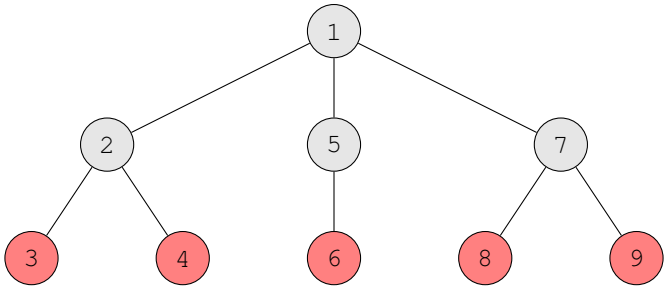
Árvores

- A **raíz** de uma árvore é o vértice que não tem pai;



Árvores

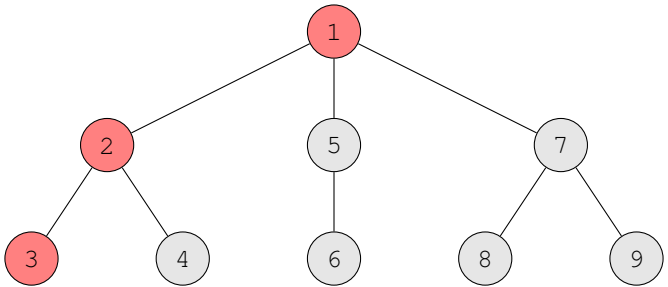
- As **folhas** de uma árvore são os vértices que não têm filhos;



centro universitário FEI

Árvores

- A **altura** de uma árvore é a distância máxima da raiz até qualquer folha;



Árvores

Utilidade

- Útil para representar dados **hierárquicos**.
- É possível representar **outras estruturas de dados** através de árvores (como listas!).
- Operações de busca são **mais rápidas** em alguns tipos de árvores.

centro universitário FEI

Árvores

Desempenho

Operação	Listas	Árvores
Inserção	$O(n)$	$O(\log_b n)$
Remoção	$O(n)$	$O(\log_b n)$
Busca	$O(n)$	$O(\log_b n)$

Considerando árvore balanceada.

Árvores

Propriedades

Da árvore

- Tamanho: o número de nós na árvore
- Altura: a distância da raiz ao nó mais profundo

Do nó

- Profundidade (nível): a distância de um nó até a raiz. Nó raiz tem normalmente nível 0.

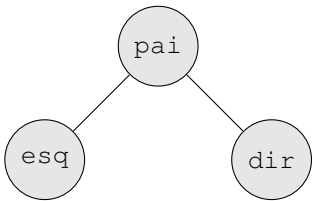
centro universitário FEI

Árvores Binárias

centro universitário FEI

Árvores

Árvore Binária



- Cada vértice possui:
  - no mínimo 0 e
  - no máximo 2 filhos
- Os filhos de um vértice são chamados de:
  - filho esquerdo e
  - filho direito

Árvores Binárias de Busca

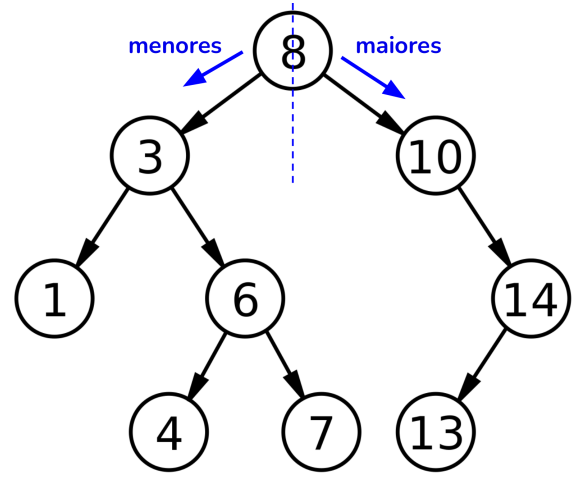
Propriedade estrutural

- Na árvore binária de busca, os valores na sub-árvore; esquerda de um nó devem ser **menores ou iguais** ao valor do próprio vértice;
- Os valores na sub-árvore direita de um nó devem ser **maiores ou iguais** aos valor do próprio vértice;
- Isso significa que a árvore possui **ordenação**.

centro universitário FEI

Árvores Binárias de Busca

Exemplo



Travessias

Árvores Binárias de Busca

Travessia em árvores

- Como as Árvores são **não-lineares**, é possível visitar seus vértices de mais de uma forma;
- Os algoritmos de **travessia** são utilizados para visitar todos os vértices de uma árvore apenas uma vez;
- Além da travessia **em ordem**, podemos percorrer uma árvore binária de outras 2 formas:
  - **pré-ordem**;
  - **pós-ordem**.

centro universitário FEI

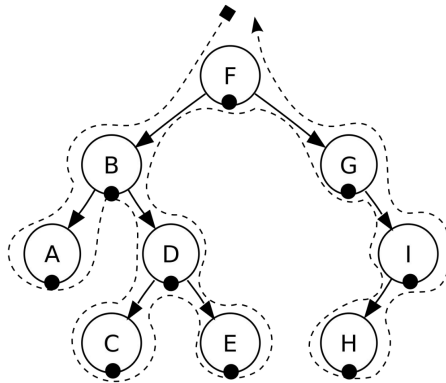
Árvores Binárias de Busca

Travessia em ordem

Esquerda-Raiz-Direita (E-R-D)

1. visita subárvore esquerda, em ordem E-R-D
2. nó raiz
3. visita subárvore direita, em ordem E-R-D

A, B, C, D, E, F, G, H, I



centro universitário FEI

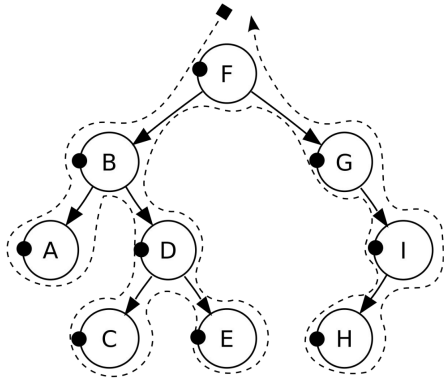
Árvores Binárias de Busca

Travessia em pré-ordem

Raiz-Esquerda-Direita (R-E-D)

1. nó raiz
2. visita subárvore esquerda, em ordem R-E-D
3. visita subárvore direita, em ordem R-E-D

F, B, A, D, C, E, G, I, H



centro universitário FEI

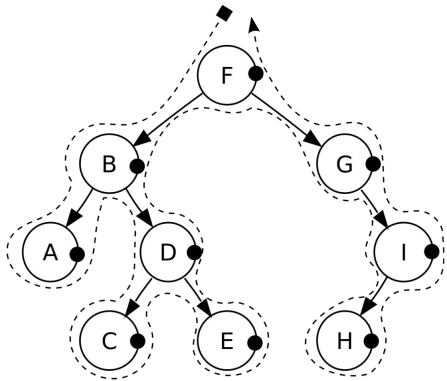
Árvores Binárias de Busca

Travessia em pós-ordem

Esquerda-Direita-Raiz (E-D-R)

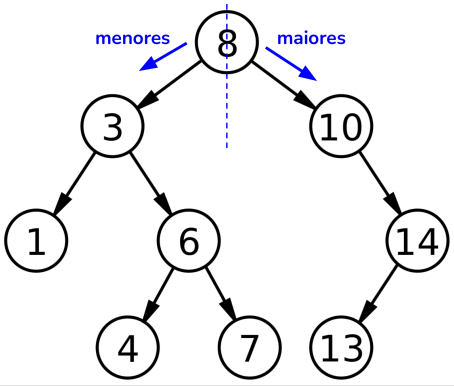
1. visita subárvore esquerda, em ordem E-D-R
2. visita subárvore direita, em ordem E-D-R
3. nó raiz

A, C, E, D, B, H, I, G, F



centro universitário FEI

Mostre o resultado das travessias



centro universitário FEI

Breve Revisão de Recursão

Recursividade

Definição

- Capacidade que uma função apresenta de **chamar a si mesma**;
- Um algoritmo recursivo é a **transformação do problema** (instância) original em **outro menor ou mais simples**, permitindo uma **solução direta**;
- A condição de parada de um algoritmo recursivo é denominada de **caso base**;
- O caso base **retorna a solução direta** para a instância que realizou a chamada, até que a solução para o problema original seja obtida.

centro universitário FEI

Recursividade

A série de Fibonacci

- A sequência, ou série, de **Fibonacci** tem os valores iniciais 0 e 1, o que se caracteriza como o caso base da recursão;
- Os demais valores são obtidos a partir da **soma dos dois anteriores**, essa é a regra de formação.

$$Fib(n) = \begin{cases} 0 & \text{se } n = 1 \\ 1 & \text{se } n = 2 \\ Fib(n-1) + Fib(n-2) & \text{se } n > 2 \end{cases}$$

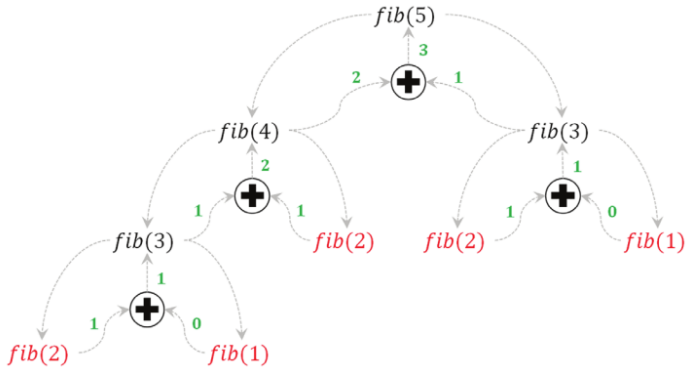
[1] Sequência de Fibonacci: definição recursiva com indexação ordinal para facilitar o aprendizado de recursão em estruturas de dados.

- O **algoritmo recursivo** que retorna o valor da n-ésima posição da série de Fibonacci:

$FIB(n)$
1 se $n = 1$ ou $n = 2$ então
2 retornar $n - 1$
3 senão
4 retornar $FIB(n-1) + FIB(n-2)$

Recursividade

Chamadas recursivas do algoritmo de Fibonacci



Descreva em pseudocódigo algoritmos recursivos para as travessias

Operações sobre Árvore Binária

Árvores Binárias de Busca

Operação de inserção em uma árvore binária

- A função de inserção recebe dois parâmetros: a raiz da árvore e um valor a ser inserido;
- Passos:
  1. Se a árvore for vazia, crie um novo vértice com o valor e torne-o a raiz;
  2. Inicialize dois ponteiros: atual (começando na raiz) e pai (inicialmente nulo);
  3. Percorra a árvore até encontrar posição vazia:
     - Se valor < atual.valor, vá para a esquerda
     - Se valor > atual.valor, vá para a direita
     - Se valor = atual.valor, defina política (substituir ou ignorar)
  4. Atualize pai = atual antes de mover atual;
  5. Quando atual = NULL, crie novo vértice e conecte como filho do pai na direção apropriada.

centro universitário FEI

Árvores Binárias de Busca

Operação de remoção em uma árvore binária

- A função de remoção recebe dois parâmetros: a raiz da árvore e um valor a ser removido;
- Passos:
  1. Busque o vértice que contém o valor (retorne se não encontrar);
  2. Analise o número de filhos do vértice encontrado:
  3. **Caso 0 filhos:** Remova o vértice e ajuste ponteiro do pai para NULL;
  4. **Caso 1 filho:** Conecte o filho diretamente ao pai do vértice removido;
  5. **Caso 2 filhos:** Encontre o predecessor in-order (maior valor da subárvore esquerda), substitua o valor do vértice a ser removido pelo valor do predecessor, e chame recursivamente a função de remoção para eliminar o predecessor.

centro universitário FEI

Mostre inserção e remoção considerando os seguintes valores na ordem em que aparecem:

$$5, 3, 7, 1, 8, 4, 6, 2, 9$$

centro universitário FEI

Estrutura de Dados (CC4652)

Aula 10 – Árvores Binárias de Busca

Prof. Luciano Rossi
Prof. Leonardo Anjoletto Ferreira
Prof. Flavio Tonidandel
Prof. Fabio Suim

Ciência da Computação
Centro Universitário FEI

2º Semestre de 2025