INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SUL-RIO-GRANDENSE
Campus Passo Fundo

Ciência da
Computação

ESTRUTURA DE DADOS II

Prof. Adilso Nunes de Souza

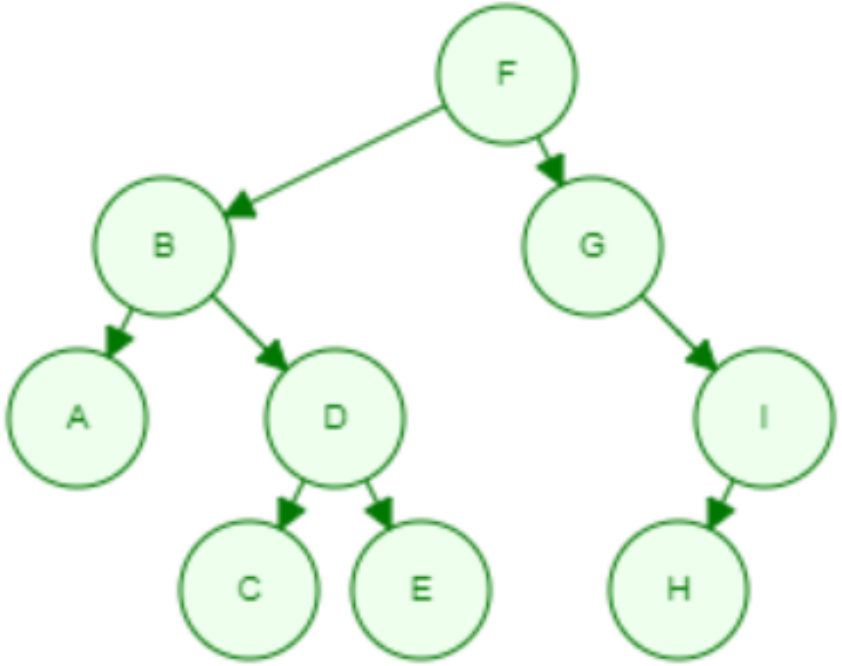
BUSCA EM ÁRVORE BINÁRIA

- Qualquer operação de busca deve ter um critério claro para percorrer a árvore, **sabendo sempre qual sua estrutura de criação**, e ir "descendo", ora para esquerda, ora para direita, até encontrar o dado.
- Às vezes o dado que procuramos não está na árvore. Percebemos isso, porque em um certo momento, ao tentar descer mais um nó, simplesmente chegamos ao final da árvore.

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SUL-RIO-GRANDENSE
Campus Passo Fundo

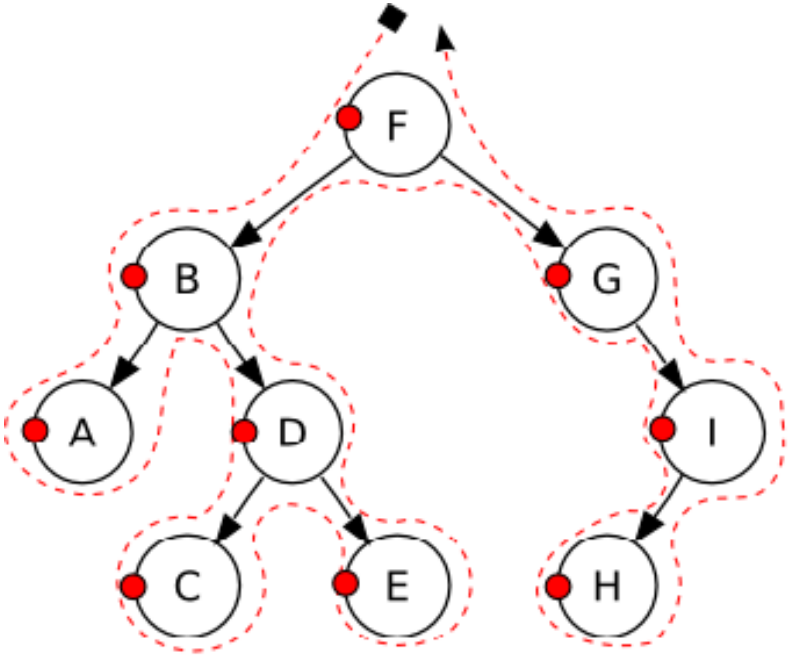
BUSCA EM ÁRVORE BINÁRIA

- Localizar o valor E na árvore



ÓRDEM DE PERCURSO

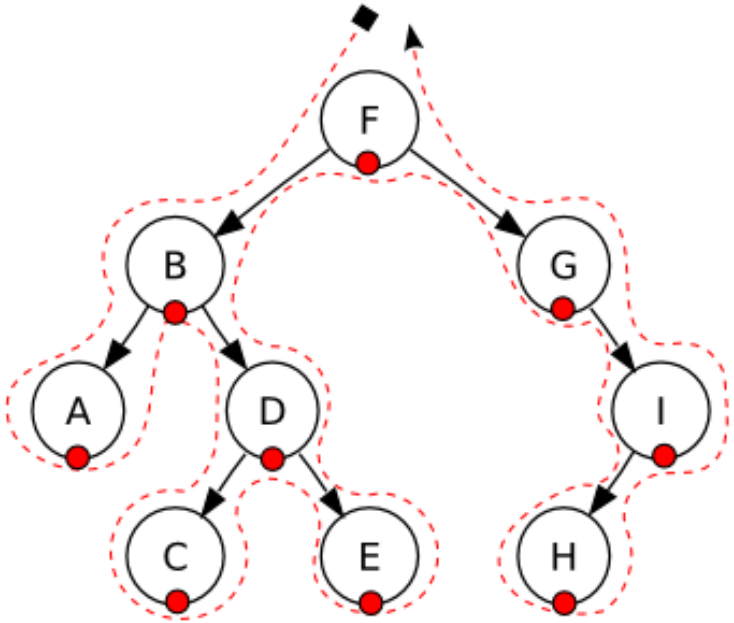
- **pré-ordem:**
  - trata raiz, percorre sae, percorre sad



<F<B<A<><><>><D<C<><><>><E<><><>>>><G<><I<H<><><>><><>>>>

ÓRDEM DE PERCURSO

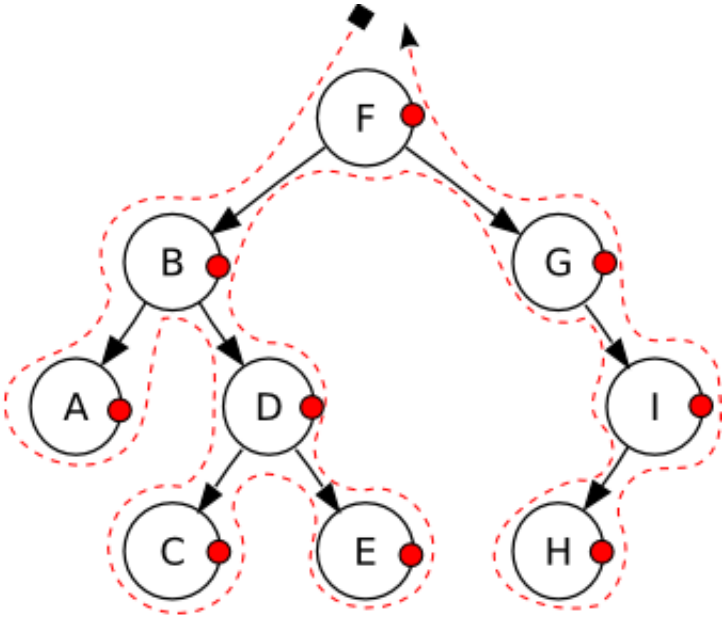
- **ordem simétrica ou in-ordem:**
  - percorre sae, trata raiz, percorre sad



<<<<>A<>>B<<<>C<>>D<<>E<>>>>F<<>G<<<>H<>>I<>>>>

ÓRDEM DE PERCURSO

- **pós-ordem:**
  - percorre sae, percorre sad, trata raiz



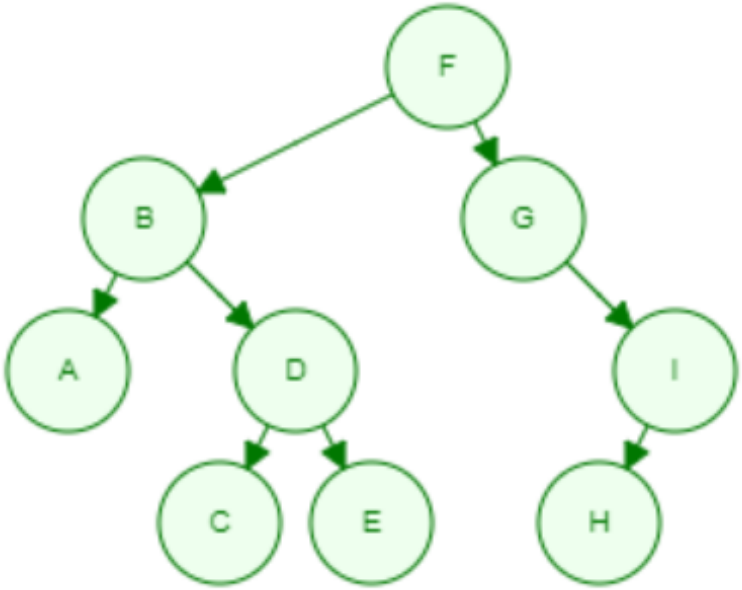
<<<<><>A><<<><>C><<><>E>D>B><<><<<><>H><>I>G>F>

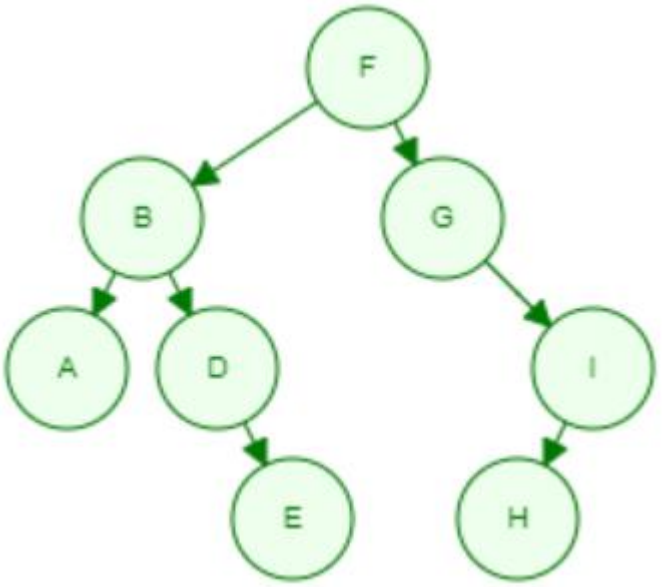
EXCLUSÃO

- Ao se retirar um dado da árvore, três situações podem ocorrer :
  - ➢ O nó removido é um nó folha (não tem filhos) : então basta removê-lo, colocando um NULO em seu lugar.

EXCLUSÃO

- Removendo o nó folha C:



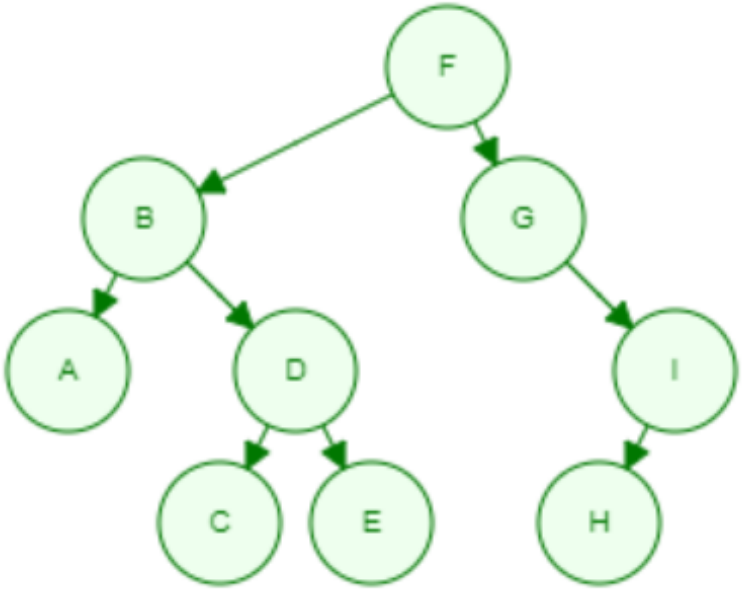


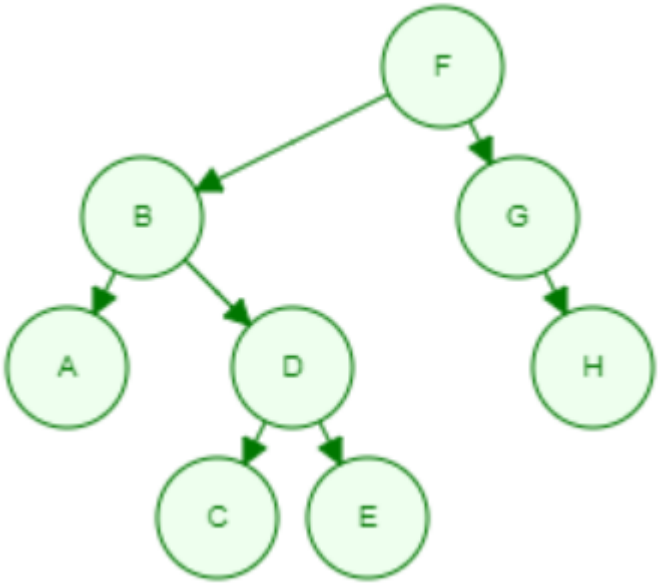
EXCLUSÃO

- O nó removido tem um filho (que pode ser à esquerda, ou à direita) : nesse caso, o filho toma o lugar do nó removido.

EXCLUSÃO

- Removendo o nó I:





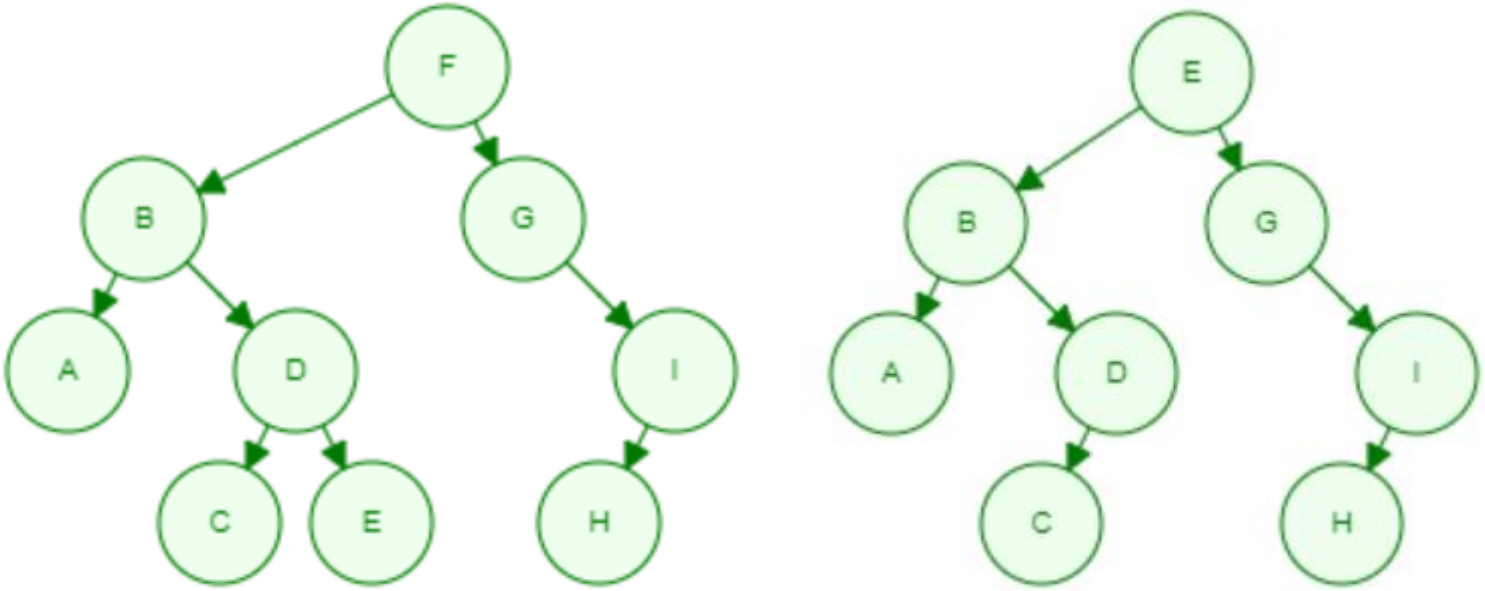
EXCLUSÃO

- O nó removido tem dois filhos: Se isso ocorrer, outro nó deve ser escolhido para substituir o nó removido.
- Existem duas soluções igualmente corretas e viáveis :
- ✓ **Podemos pegar o nó mais à direita na sub-árvore esquerda de quem foi removido;***
- ✓ Ou pegar o nó mais à esquerda na sub-árvore direita do nó removido para substituí-lo.

*Está será a solução adotada

EXCLUSÃO

- Removendo o nó F:



PROBLEMAS

- As ações de inserção e remoção em árvores binárias não garantem que a árvore gerada a cada passo esteja balanceada.
- A eficiência da busca em uma árvore binária depende do seu balanceamento.
- Quanto maior a altura da árvore, maior será o custo da operação de busca.

SOLUÇÃO

- Modificar as operações de inserção e remoção da árvore para balancear a árvore a cada nova inserção ou remoção.
- Existem diferentes tipos de árvores balanceadas, conforme a aplicação computacional e o algoritmo criado:

TIPOS DE ÁRVORES BALANCEADAS

- Árvore AVL
- Árvore Vermelho-Preto (Red-Black)
- Árvore 2-3
- Árvore 2-3-4
- Árvore B, B+ e B*

ATIVIDADES

- Criar as respectivas funções:
  - Encontrar a altura da sub-árvore esquerda e direita
  - Encontrar a altura da árvore
  - Limpar todos os nós da árvore, liberando as respectivas memórias.
  - Dado um determinado valor, se o mesmo estiver contido na árvore encontrar o nível que ele se encontra.

REFERÊNCIAS

- PEREIRA, Silvio do Lago. Estrutura de Dados Fundamentais: Conceitos e Aplicações, 12. Ed. São Paulo, Érica, 2008.
- BACKES, André Ricardo, Estrutura de dados descomplicada: em linguagem C, 1 Ed. – Rio de Janeiro: Elsevier, 2016.
- SENGER, H., Notas de Aula, Universidade de São Judas Tadeu, 1999.
- WALDEMAR Celes, Renato Cerqueira, José Lucas Rangel, Introdução a Estruturas de Dados, Editora Campus (2004).
- VELOSO, Paulo. SANTOS, Celso dos. AZEVEDO, Paulo. FURTADO, Antonio. Estrutura de dados. Rio de Janeiro: Ed. Elsevier, 1983 27ª reimpressão.