

# Aula 15 - Árvores Binárias de Busca

Estruturas de Dados 2018/1

Prof. Diego Furtado Silva

# Tabela de símbolos

Abstração do par chave-valor

- Insere um valor associado a uma chave
- Dada uma chave, **busca** o valor correspondente

Ex: *DNS lookup* (Domain Name System)

URL	Endereço IP
ufscar.br	200.136.207.48
dc.ufscar.br	200.18.99.34
ava.ead.ufscar.br	200.133.224.2

# Tabela de símbolos

Aplicação	Chave	Valor
Dicionário	Palavra	Definição
Índice remissivo	Termo	Lista de número de páginas
Spotify e afins	Nome da música	Áudio relativo à música buscada
Compilador	Nome da variável	Tipo e valor
Sistemas de arquivos	Nome do arquivo	Localização do arquivo no disco
Busca web	Termo	Páginas relacionadas à busca

# Tabela de símbolos

Aplicação	Chave	Valor
Dicionário	Palavra	Definição
Índice remissivo	Termo	Lista de número de páginas
Spotify e afins	Nome da música	Áudio relativo à música buscada
Compilador	Nome da variável	Tipo e valor
Sistemas de arquivos	Nome do arquivo	Localização do arquivo no disco
Busca web	Termo	Páginas relacionadas à busca

Vocês sabem implementar uma tabela de símbolos?

# Relembrando - Busca em vetor

**Busca linear:** varrer o vetor da primeira à (potencialmente) última posição, comparando o item buscado com o item armazenado na posição atual da varredura

Exemplo no quadro

# Relembrando - Busca em vetor

**Busca linear:** varrer o vetor da primeira à (potencialmente) última posição, comparando o item buscado com o item armazenado na posição atual da varredura

Busca em  $O(n)$

Procedimento similar para listas ligadas

# Relembrando - Busca em vetor

Busca **binária**: Dado um vetor ordenado, utilizamos a estratégia de reduzir para conquistar.

Exemplo no quadro

# Relembrando - Busca em vetor

Busca **binária**: Dado um vetor ordenado, utilizamos a estratégia de reduzir para conquistar.

Busca em  **$O(\log n)$**

Não aplicável a listas ligadas. Por quê?



# Relembrando - Busca em vetor

Busca **binária**: Dado um vetor ordenado, utilizamos a estratégia de reduzir para conquistar.

Busca em  **$O(\log n)$**

E se não estiver ordenado?

# Relembrando - Busca em vetor

E se eu tiver que fazer algumas alterações?

**SPOILER ALERT!**

Vamos voltar no exemplo do cadastro de mutantes da Escola para Jovens Superdotados, agora com o Deadpool



# Relembrando - Busca em vetor

Deadpool vira um membro (trainee) dos X-Men

- Incluimos ele no cadastro

Quando ele morre, ele sai do cadastro de atuantes

- Removemos ele desse cadastro

Podemos incluir e remover muitos itens durante a execução da minha aplicação. O que acontece com a eficiência das operações?

# Pergunta da aula

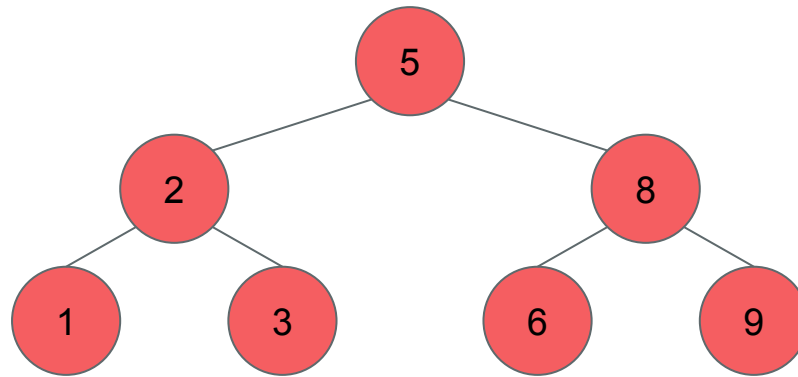
Como melhorar a implementação de uma tabela de símbolos usando árvores binárias?

# Árvores Binárias de Busca (ABB)

Uma ABB é uma AB tal que

- Os nós pertencentes à sub-árvore **esquerda** de um nó possuem chaves **menores** que de tal (nó) raiz
- Os nós pertencentes à sub-árvore **direita** de um nó possuem chaves **maiores** que de tal (nó) raiz

# Árvores Binárias de Busca (ABB)



# Árvores Binárias de Busca (ABB)

Principais operações de uma ABB

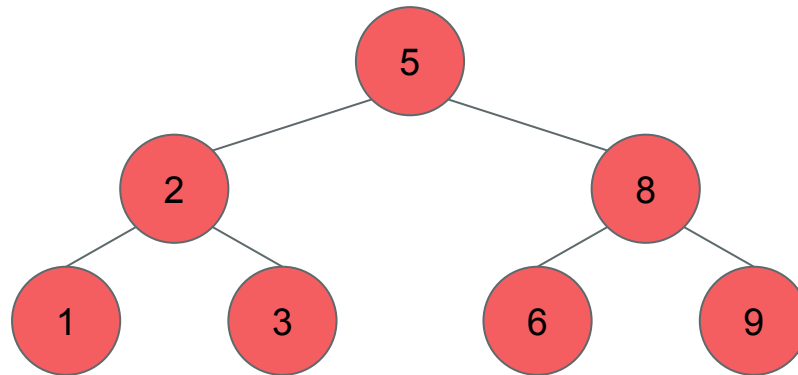
- Pesquisar
- Inserir
- Remover

Também podemos realizar percursos na ABB.

O que acontece se utilizarmos os percursos já apresentados (pré-, em e pós-ordem)?

# ABB - Pesquisa

Sugestões?

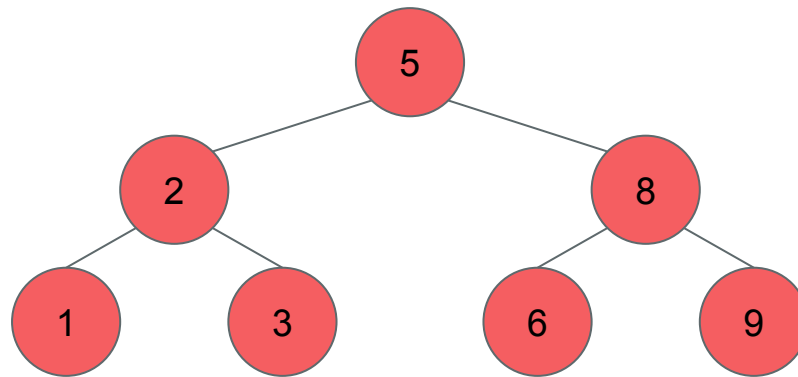




# ABB - Pesquisa

1. Iniciar a busca no nó raiz
2. Se a chave do nó é igual à pesquisada, retorna valor. Senão
  - a. Se  $\text{chave}(\text{pesquisada}) < \text{chave}(\text{nó})$ , assume a subárvore da esquerda como novo espaço de busca e retorna para o passo 2.
  - b. Se  $\text{chave}(\text{pesquisada}) > \text{chave}(\text{nó})$ , assume a subárvore da direita como novo espaço de busca e retorna para o passo 2.
  - c. Se nó é NULO, a chave não foi encontrada.

# ABB - Pesquisa



Na lousa: buscar chaves 5, 3, 8 e 7

# ABB - Inserção

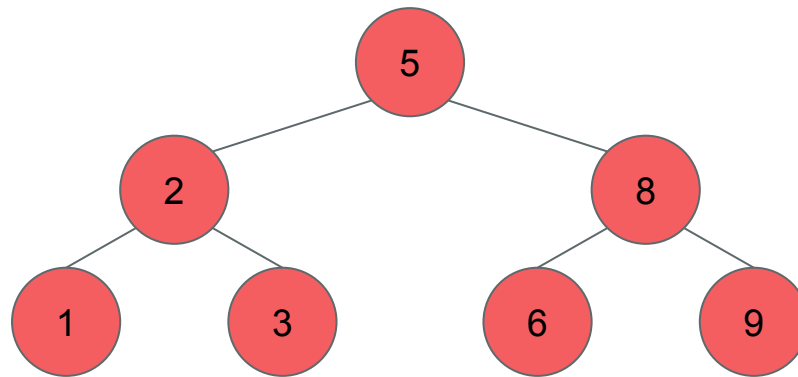
Sugestões?

# ABB - Inserção

A inserção é sempre feita nas folhas. Primeiro, encontramos a posição do novo nó folha.

1. Inicia a busca da posição no nó raiz
2. Verifica se a chave a ser inserida é maior ou menor que a chave do nó analisado
  - a. Caso seja maior e o filho à direita for nulo, insere à direita. Se não for nulo, segue para o filho da direita e volta para o passo 2.
  - b. Similar à esquerda, caso seja menor.

# ABB - Inserção



Na lousa: inserir chaves 4 e 7

# ABB - Inserção

Exercício: Inserir as chaves (a partir de uma ABB vazia)

{17,99,13,1,3,100,400}

# ABB - Remoção

Sugestões?

# ABB - Remoção

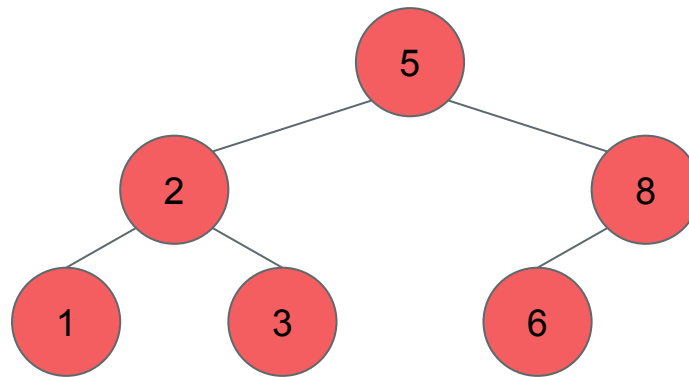
Casos a verificar:

1. O nó a ser removido é um nó folha

Maravilha! Podemos tirar o nó sem nenhum problema

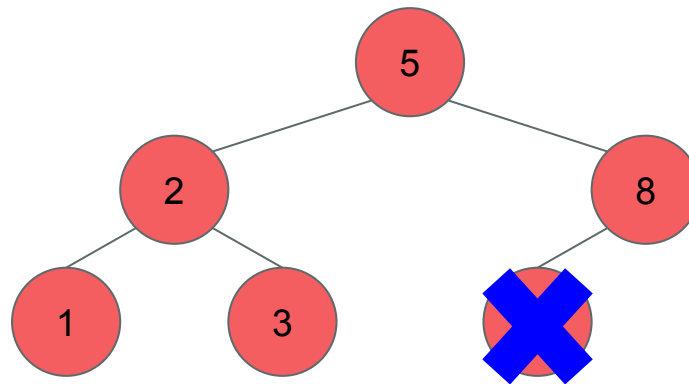


# ABB - Remoção - caso 1



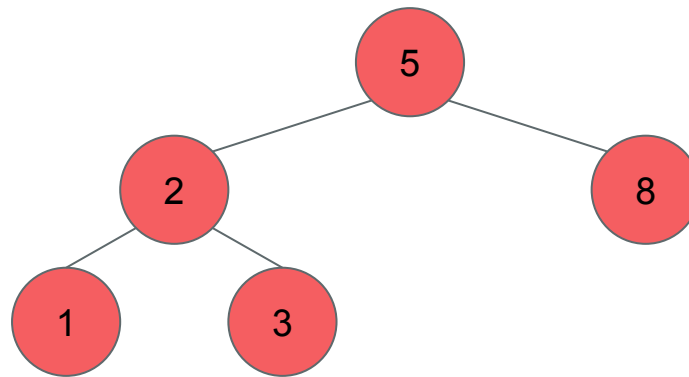
Remover a chave 6

# ABB - Remoção - caso 1



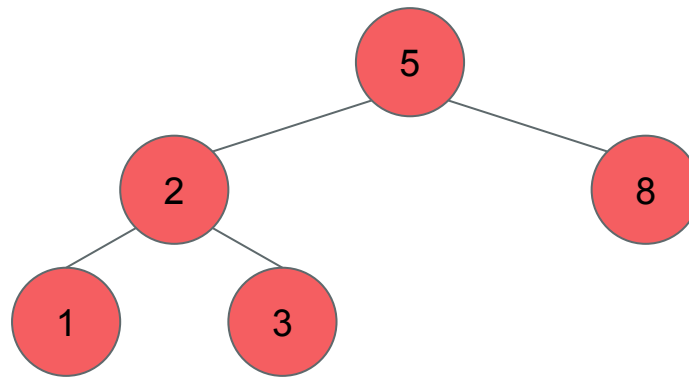
Remover a chave 6

# ABB - Remoção - caso 1



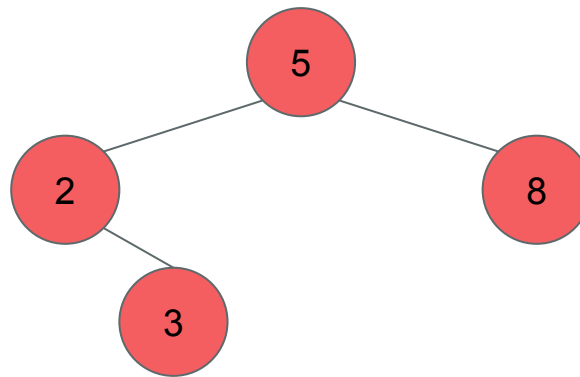
Remover a chave 6 ✓ ☐

# ABB - Remoção - caso 1



Remover a chave 1

# ABB - Remoção - caso 1



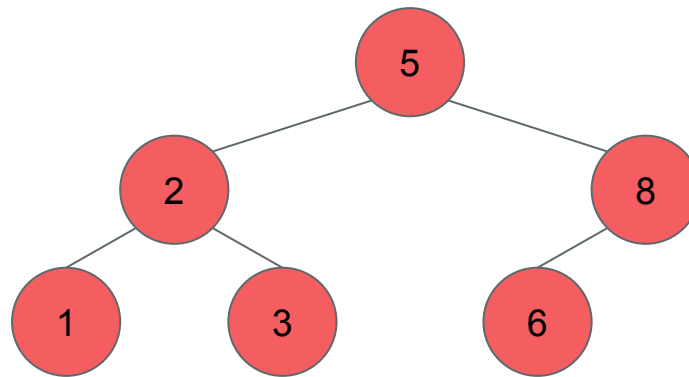
Remover a chave 1 ✓ ☐

# ABB - Remoção

Casos a verificar:

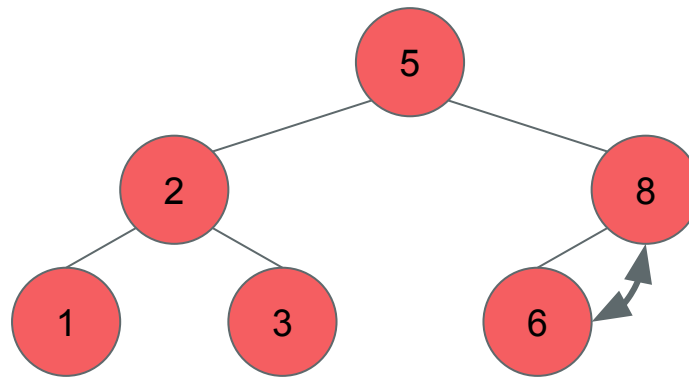
1. O nó a ser removido é um nó folha  
Maravilha! Podemos tirar o nó sem nenhum problema
2. O nó possui uma subárvore (ou direita ou esquerda)  
A raiz da subárvore pode “assumir o posto” do nó removido

## ABB - Remoção - caso 2



Remover a chave 8

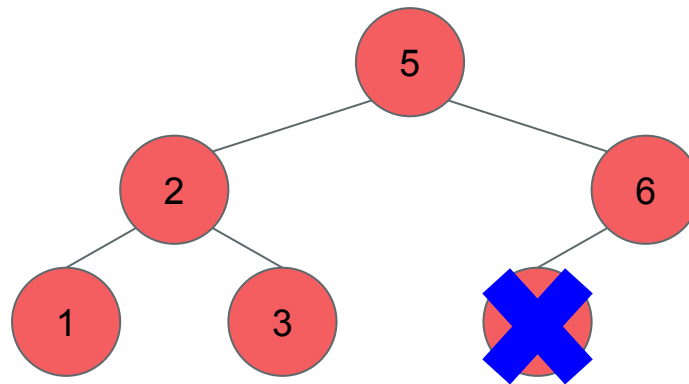
## ABB - Remoção - caso 2



Remover a chave 8

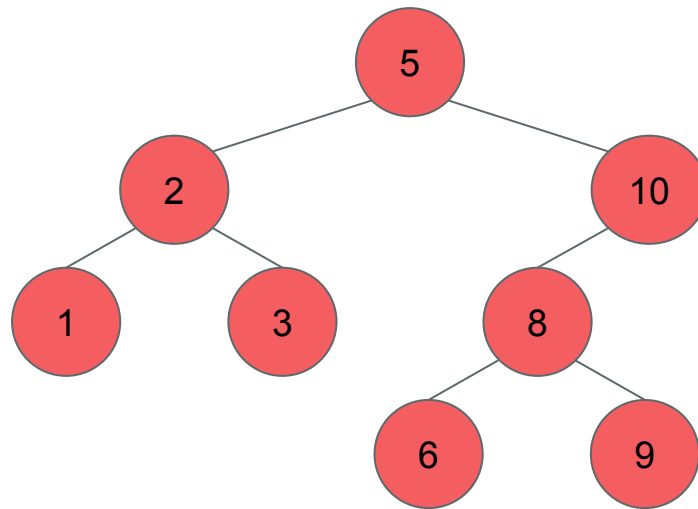


## ABB - Remoção - caso 2



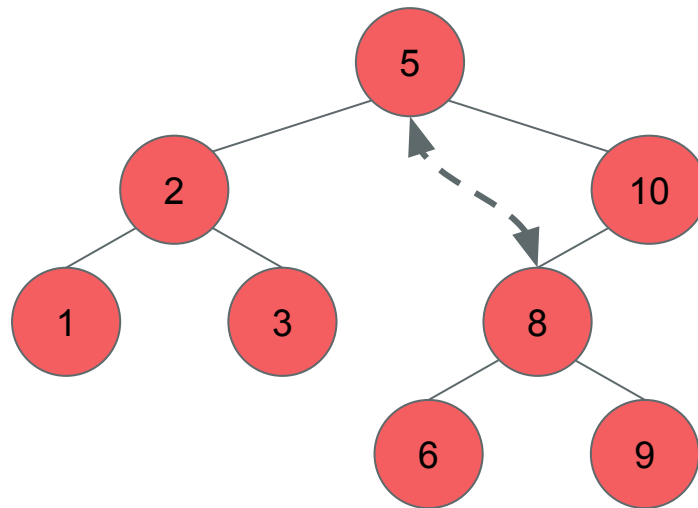
Remover a chave 8 ✓ ☐

## ABB - Remoção - caso 2



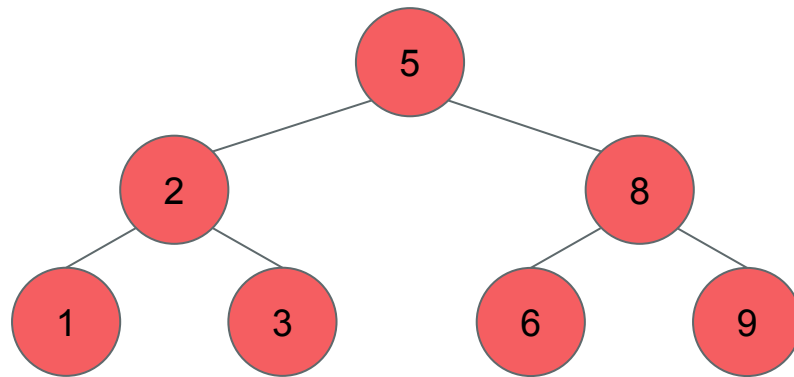
Remover a chave 10

## ABB - Remoção - caso 2



Remover a chave 10

## ABB - Remoção - caso 2



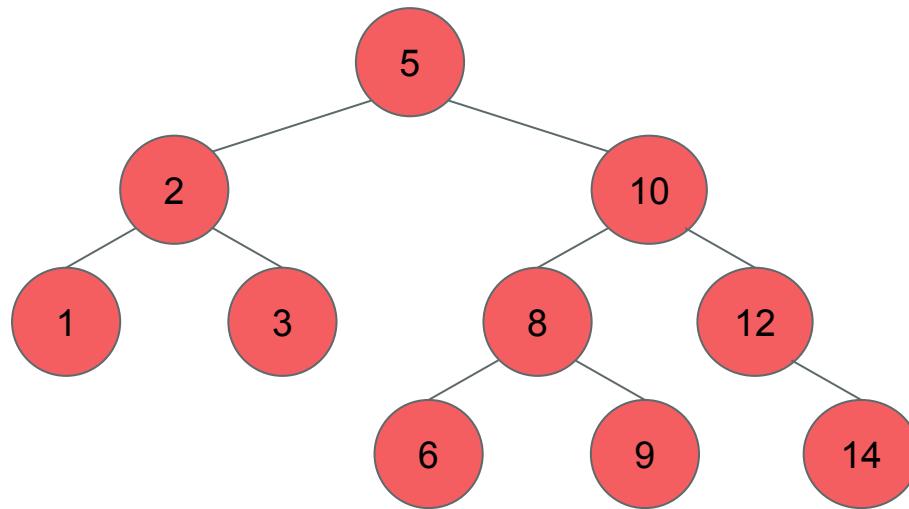
Remover a chave 10 ✓ ☐

# ABB - Remoção

Casos a verificar:

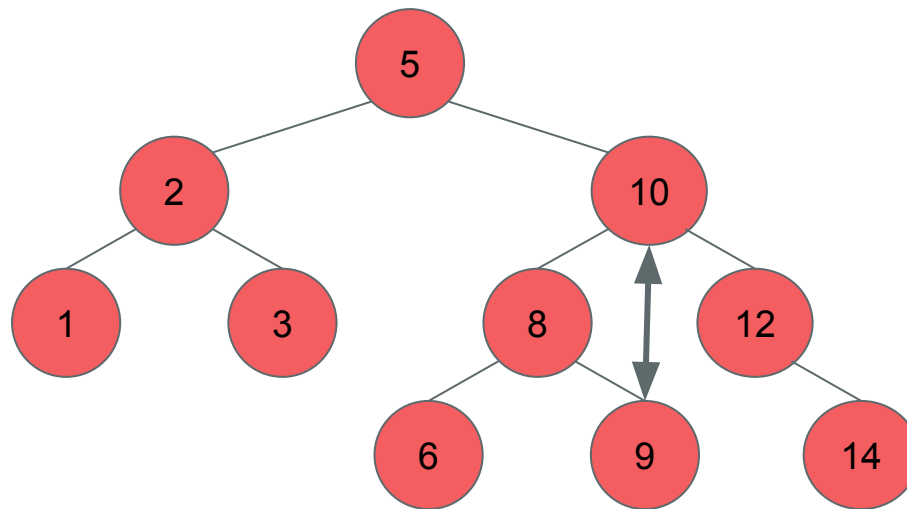
1. O nó a ser removido é um nó folha  
Maravilha! Podemos tirar o nó sem nenhum problema
2. O nó possui uma subárvore (ou direita ou esquerda)  
A raiz da subárvore pode “assumir o posto” do nó removido
3. O nó possui as duas subárvores  
Substituir pelo mais à direita do filho à esquerda (ou mais à esquerda do filho à direita). Qual a relação com percurso em ordem?

## ABB - Remoção - caso 3



Remover a chave 10

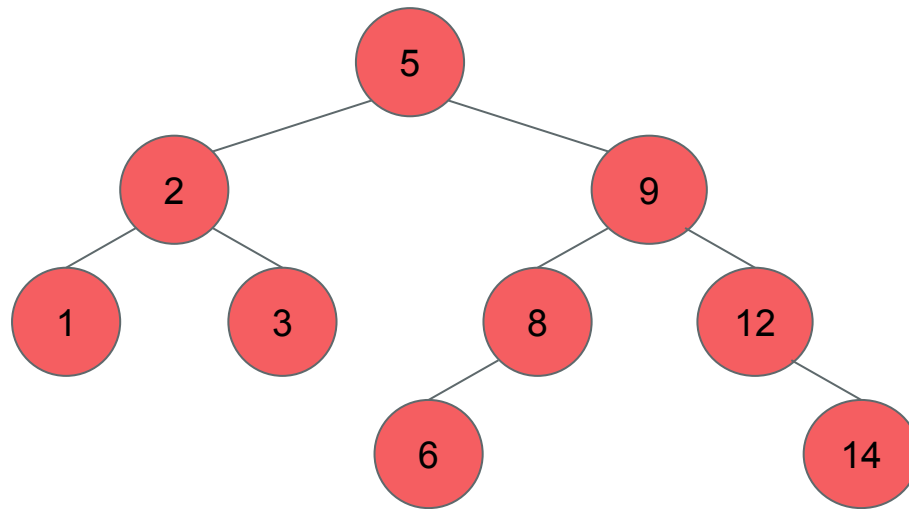
## ABB - Remoção - caso 3



Remover a chave 10

Mais à direita do filho à esquerda = 9

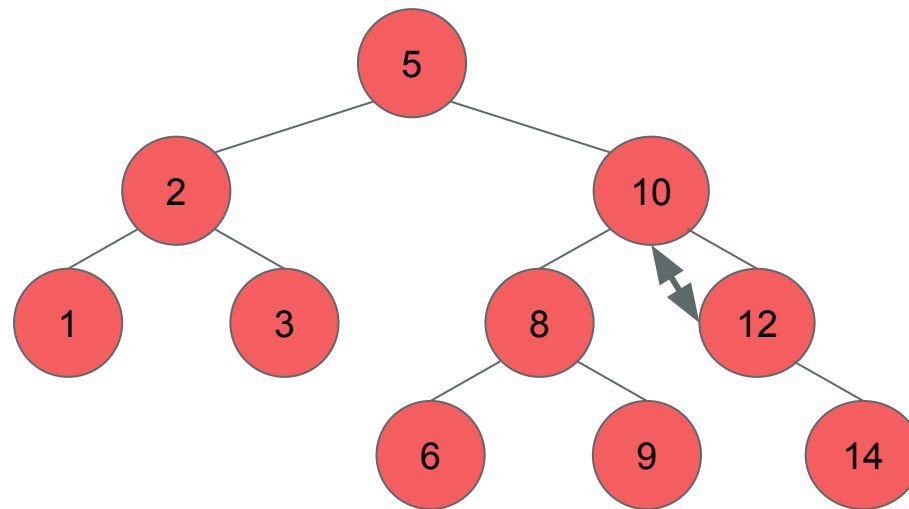
## ABB - Remoção - caso 3



Remover a chave 10 ✓ ☐



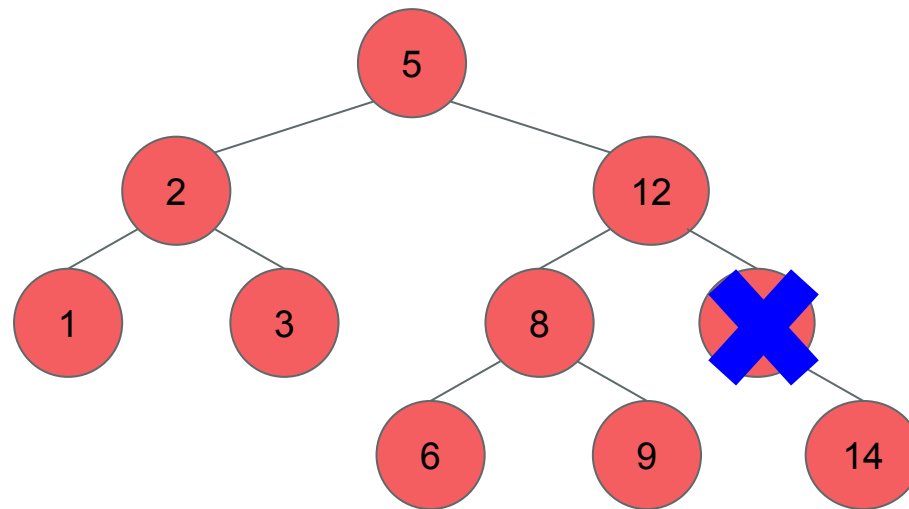
## ABB - Remoção - caso 3



Remover a chave 10

[OU] Mais à esquerda do filho à direita = 12

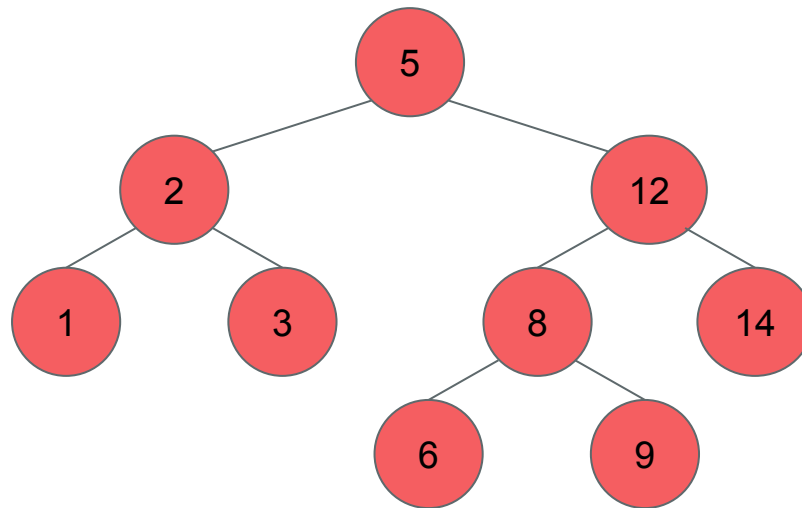
## ABB - Remoção - caso 3



Remover a chave 10

[OU] Mais à esquerda do filho à direita = 12

## ABB - Remoção - caso 3



Remover a chave 10 ✓ ☐

# ABB - Exercício

- Fazer inserções de números aleatórios em uma ABB.
- Junto a isso, depois de algumas inserções, fazer remoções aleatórias.

Como fica a geometria da árvore?

# ABB - Próxima aula

Complexidade (melhor e pior caso)  
e implementação



"You know about binary search trees?" "Say no more"