



UNIVERSIDADE FEDERAL DE ITAJUBÁ

Algoritmos e Estrutura de Dados I

COM 111

Árvore Binária de Pesquisa

Vanessa Cristina Oliveira de Souza

Operações em Árvores Binárias



TAD ABP

- ▶ As principais operações sobre as árvores binárias são:
 - ▶ Inserir um elemento
 - ▶ **Excluir um elemento**
 - ▶ Buscar um elemento
 - ▶ Encontrar o maior
 - ▶ Encontrar o menor
 - ▶ Buscar o elemento sucessor e o predecessor
 - ▶ Percorrimento
 - ▶ Em pré-ordem
 - ▶ Em ordem
 - ▶ Em pós-ordem





Remoção em ABP

- ▶ A remoção de um nó da árvore binária de pesquisa depende da quantidade de filhos que ele possui.
- ▶ Sendo assim:
 - ▶ Nó folha (sem filhos)
 - ▶ Nó com um único filho
 - ▶ Nó com dois filhos



Remoção em ABP

► Nó folha (Grau 0)

► A remoção de um nó folha é a mais simples.

1. Busca o nó a ser removido
2. Faz seu pai apontar pra NULL
3. Remove o nó



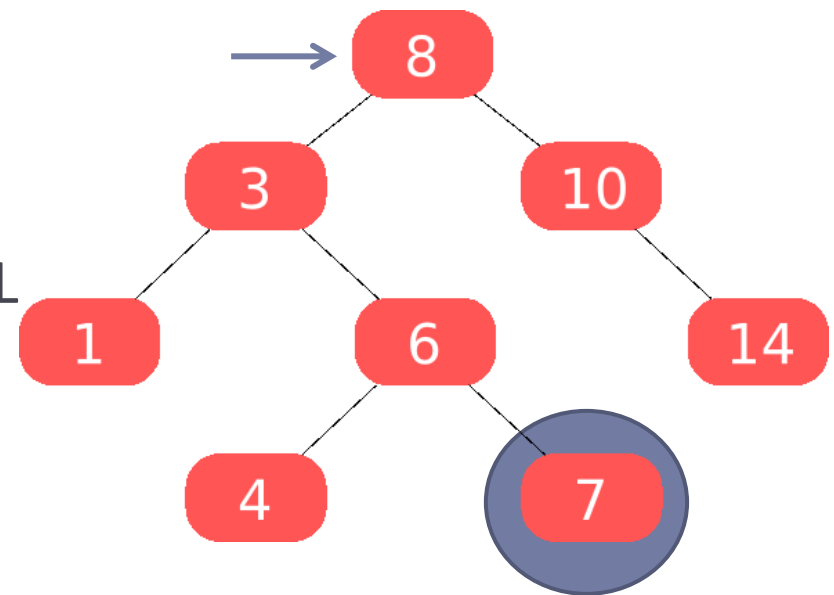


Remoção em ABP

► Nó folha

► Remover o nó 7

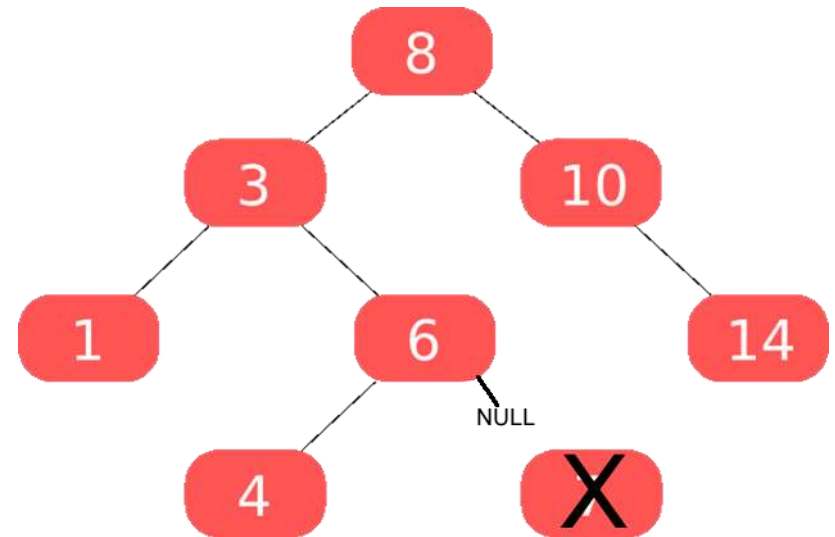
1. Busca o nó a ser removido
2. Faz seu pai apontar pra NULL
3. Remove o nó





Remoção em ABP

- ▶ **Nó folha**
 - ▶ **Remover o nó 7**
 - ▶ Busca o elemento
 - ▶ Faz seu pai apontar pra NULL
 - ▶ Remove o nó





Remoção em ABP

► Nó com um filho (Grau 1)

1. Busca o nó a ser removido
2. Faz o pai do nó apontar para seu único filho
3. Remove o nó



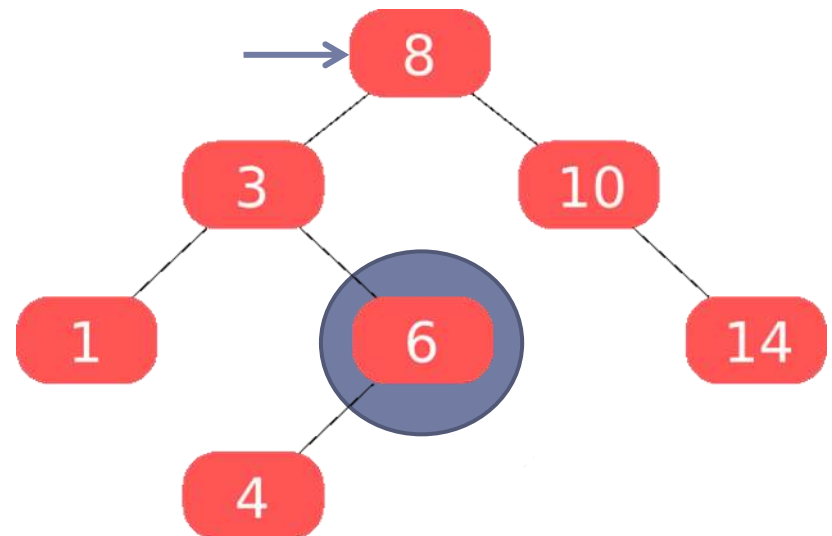


Remoção em ABP

► Nó com um filho

► Remover o nó 6

1. Busca o nó a ser removido
2. Faz o pai do nó apontar para seu único filho
3. Remove o nó



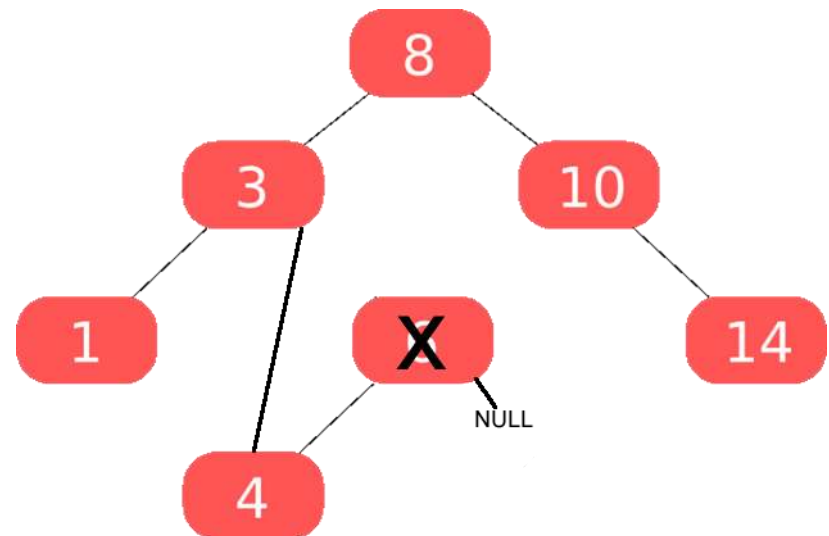


Remoção em ABP

► Nó com um filho

► Remover o nó 6

1. Busca o nó a ser removido
2. Faz o pai do nó apontar para seu único filho
3. Remove o nó





Remoção em ABP

- ▶ Nó com dois filhos (Grau 2)
 - ▶ Essa é a remoção mais complexa.
 - ▶ A remoção 'física' do nó é muito cara computacionalmente.
 - ▶ Utilizamos um algoritmo chamado '**remoção por cópia**'
 1. Busca o nó a ser removido
 2. Encontra o sucessor ou predecessor do nó
 3. Copia o valor do nó (chave + dados satélite) para o lugar do nó removido
 4. Remove o nó sucessor ou predecessor
 - ▶ Obs.: deve-se sempre escolher a priori o sucessor ou o predecessor.



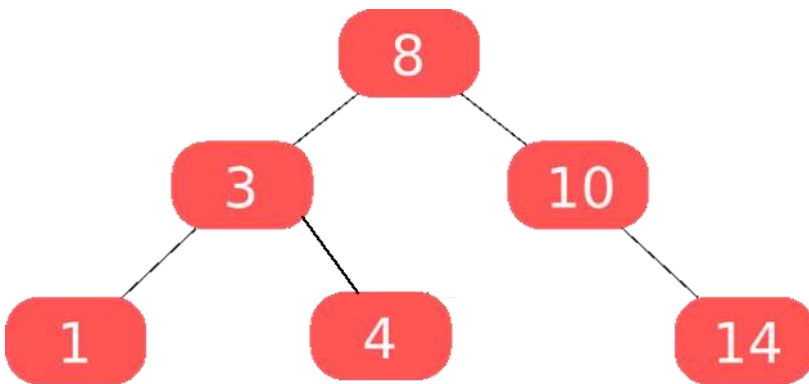


Remoção em ABP

► Nó com dois filhos

► Remover o nó 8

1. Busca o nó a ser removido
 2. Encontra o sucessor ou predecessor do nó
 3. Copia o valor do nó (chave + dados satélite) para o lugar do nó removido
 4. Remove o nó sucessor ou predecessor
- Obs.: deve-se sempre escolher a priori o sucessor ou o predecessor.



Sucessor:

- Próximo elemento maior que 8
- Nesta árvore, o sucessor é o 10
- Elemento mais à esquerda da sub-árvore à direita

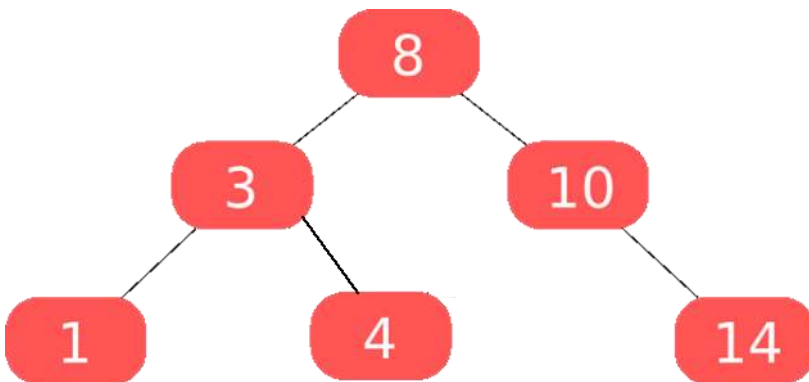


Remoção em ABP

► Nó com dois filhos

► Remover o nó 8

1. Busca o nó a ser removido
 2. Encontra o sucessor ou predecessor do nó
 3. Copia o valor do nó (chave + dados satélite) para o lugar do nó removido
 4. Remove o nó sucessor ou predecessor
- Obs.: deve-se sempre escolher a priori o sucessor ou o predecessor.



Predecessor:

- Elemento imediatamente menor que 8
- Nesta árvore, o predecessor é o 4
- Elemento mais à direita da sub-árvore à esquerda

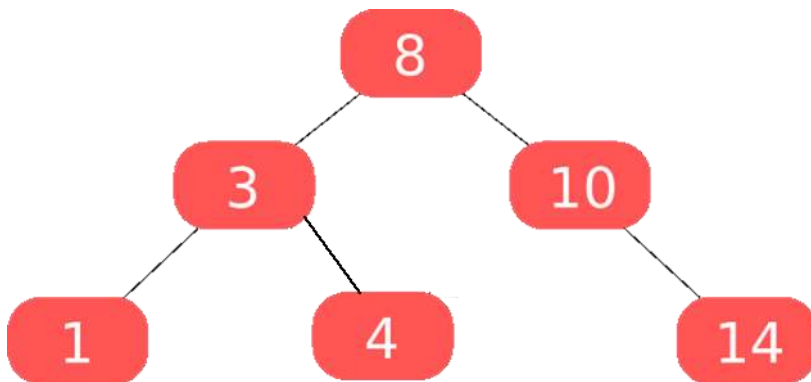


Remoção em ABP

► Nó com dois filhos

► Remover o nó 8

1. Busca o nó a ser removido
 2. Encontra o sucessor ou predecessor do nó
 3. Copia o valor do nó (chave + dados satélite) para o lugar do nó removido
 4. Remove o nó sucessor ou predecessor
- Obs.: deve-se sempre escolher a priori o sucessor ou o predecessor.



O nó sucessor ou predecessor sempre terá grau 0 ou 1!!!



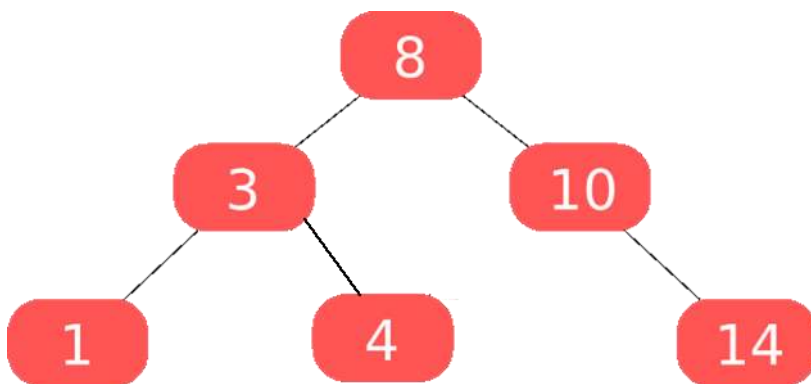


Remoção em ABP

► Nó com dois filhos

► Remover o nó 8

1. Busca o nó a ser removido
 2. Encontra o sucessor ou predecessor do nó
 3. Copia o valor do nó (chave + dados satélite) para o lugar do nó removido
 4. Remove o nó sucessor ou predecessor
- Obs.: deve-se sempre escolher a priori o sucessor ou o predecessor.



Vamos adotar o predecessor como padrão

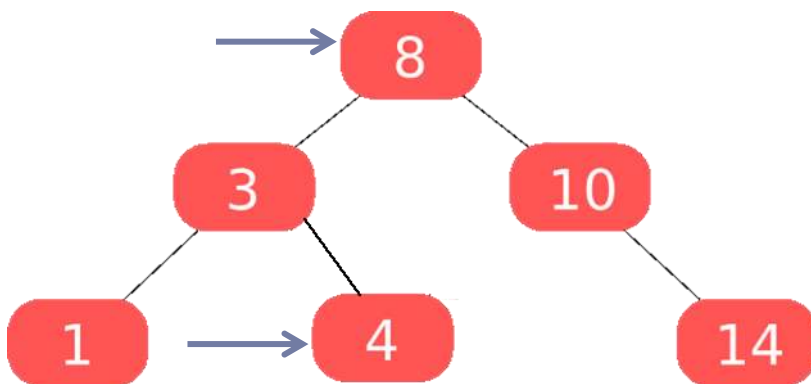


Remoção em ABP

► Nó com dois filhos

► **Remover o nó 8**

1. Busca o nó a ser removido
 2. Encontra o sucessor ou predecessor do nó
 3. Copia o valor do nó (chave + dados satélite) para o lugar do nó removido
 4. Remove o nó sucessor ou predecessor
- Obs.: deve-se sempre escolher a priori o sucessor ou o predecessor.



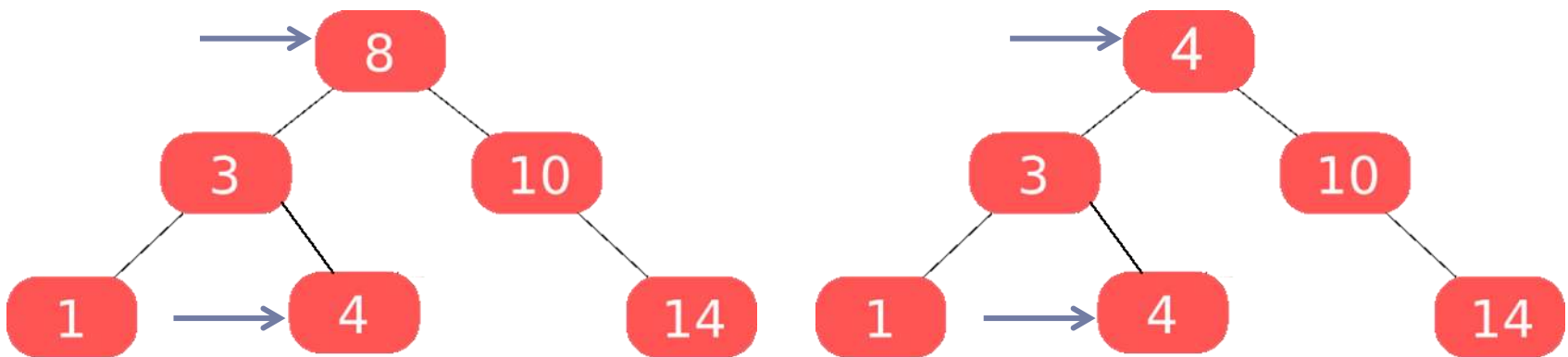


Remoção em ABP

► Nó com dois filhos

► Remover o nó 8

1. Busca o nó a ser removido
 2. Encontra o sucessor ou predecessor do nó
 3. **Copia o valor do nó (chave + dados satélite) para o lugar do nó removido**
 4. Remove o nó sucessor ou predecessor
- Obs.: deve-se sempre escolher a priori o sucessor ou o predecessor.



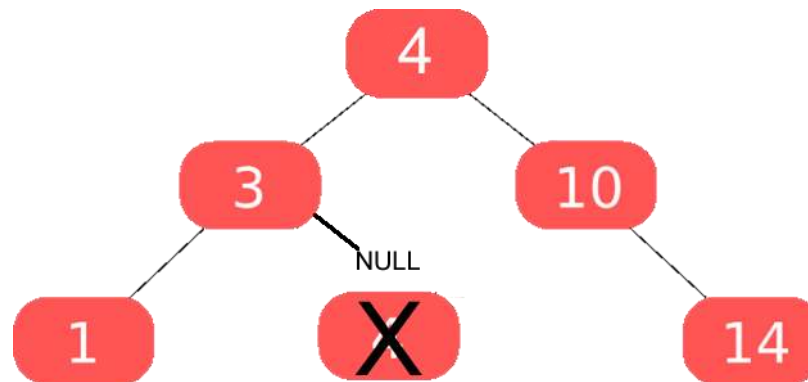


Remoção em ABP

► Nó com dois filhos

► **Remover o nó 8**

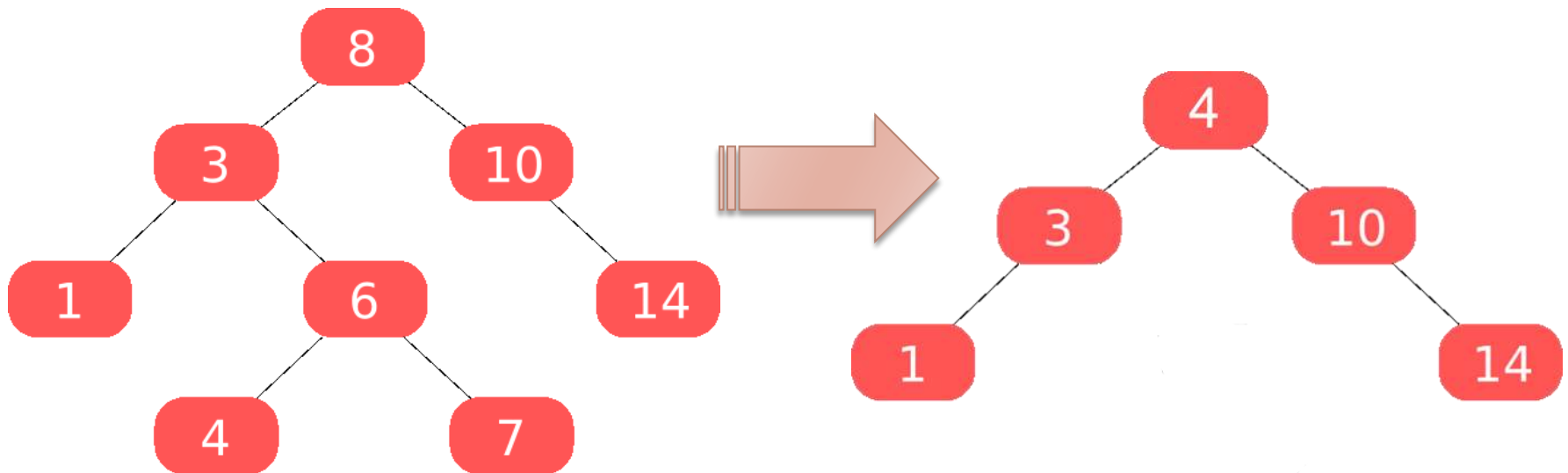
1. Busca o nó a ser removido
2. Encontra o sucessor ou predecessor do nó
3. Copia o valor do nó (chave + dados satélite) para o lugar do nó removido
4. **Remove o nó sucessor ou predecessor**
 - Obs.: deve-se sempre escolher a priori o sucessor ou o predecessor.





Remoção em ABP

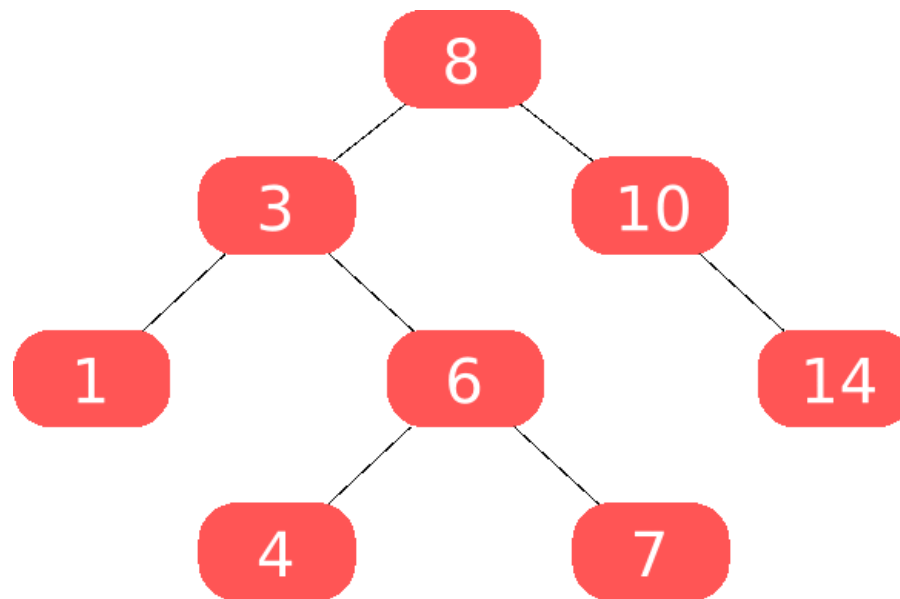
- ▶ Árvore original e árvore após a remoção dos elementos 7, 6 e 8, **nessa ordem**!





Remoção em ABP

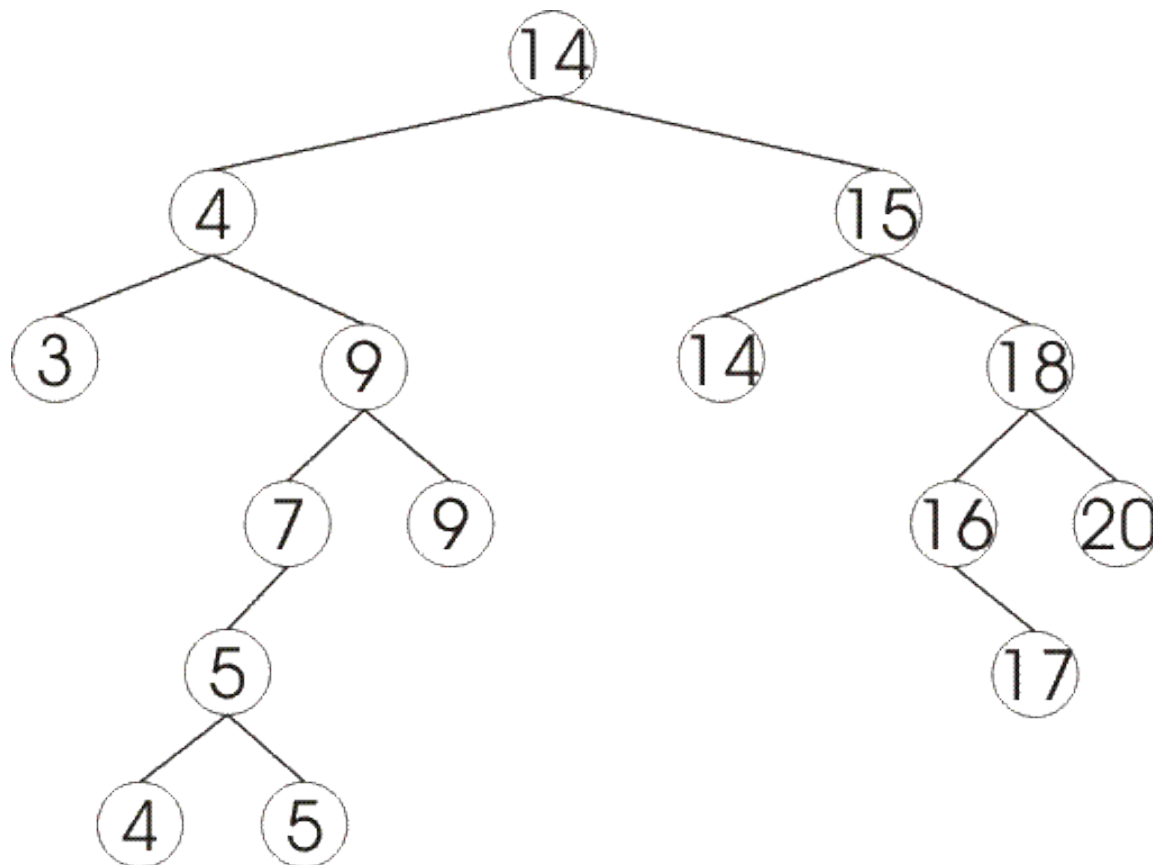
- ▶ Exercício
- ▶ Remover os elementos 6, 7 e 8 usando o sucessor.





Remoção em ABP

- E quando a árvore tem elementos repetidos?

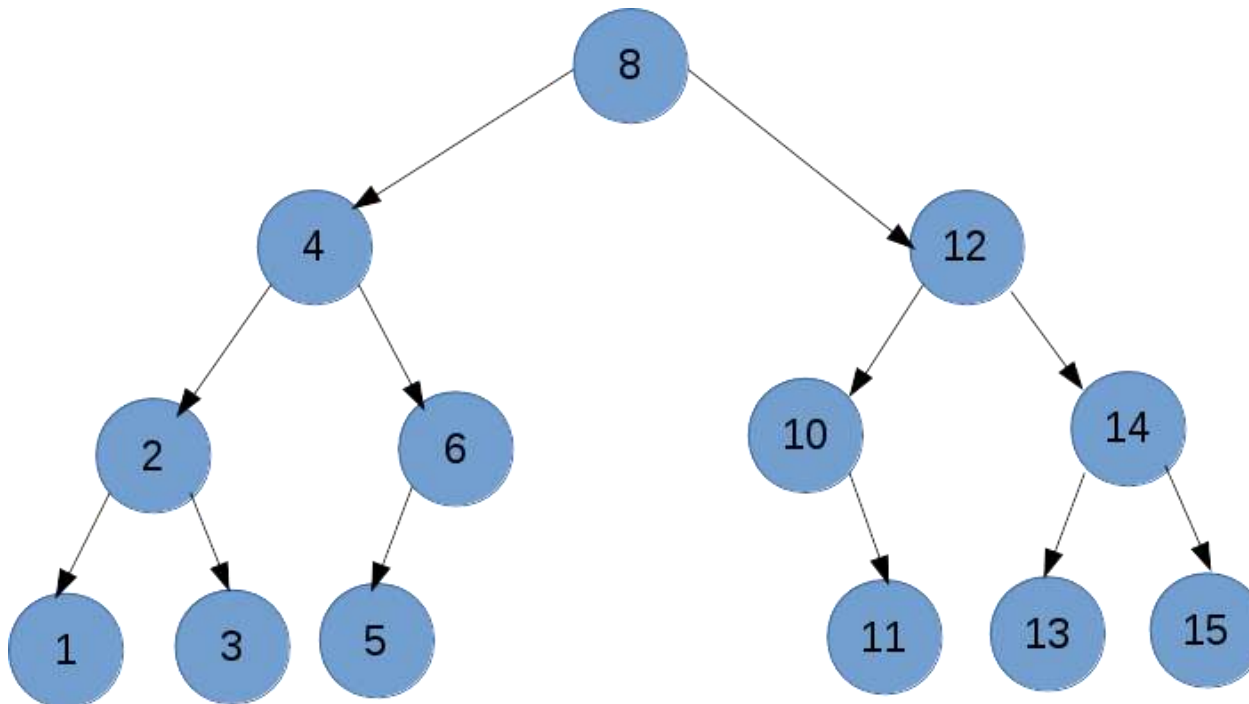




Remoção em ABP

► Exercício

- Dada a árvore binária abaixo, remova os elementos 12, 2, 4, 10 e 15, nessa ordem. Use o predecessor.





Implementação Remoção



Implementação

- ▶ `void removeElementolt(abp *A, int elemento);`
 - ▶ Encontra o nó a ser removido e envia para a função `removelt`
- ▶ `void removelt(abp *A, noArvore *no, noArvore *pai);`
 - ▶ Remove o nó da árvore