

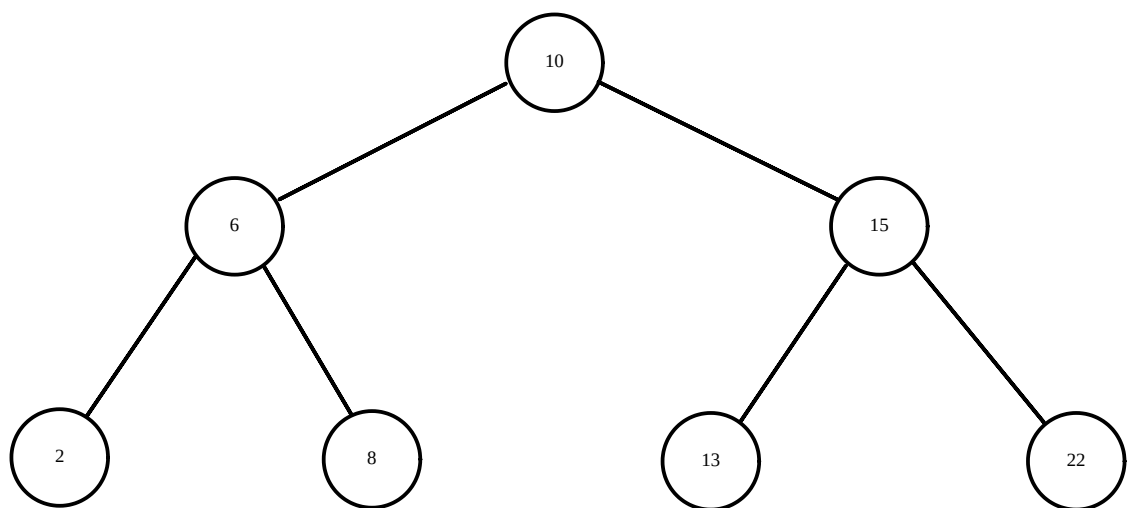
1 - Árvore Binária

▼ Matéria	Estrutura de Dados II
📎 Arquivo	

- **Árvore Binária**

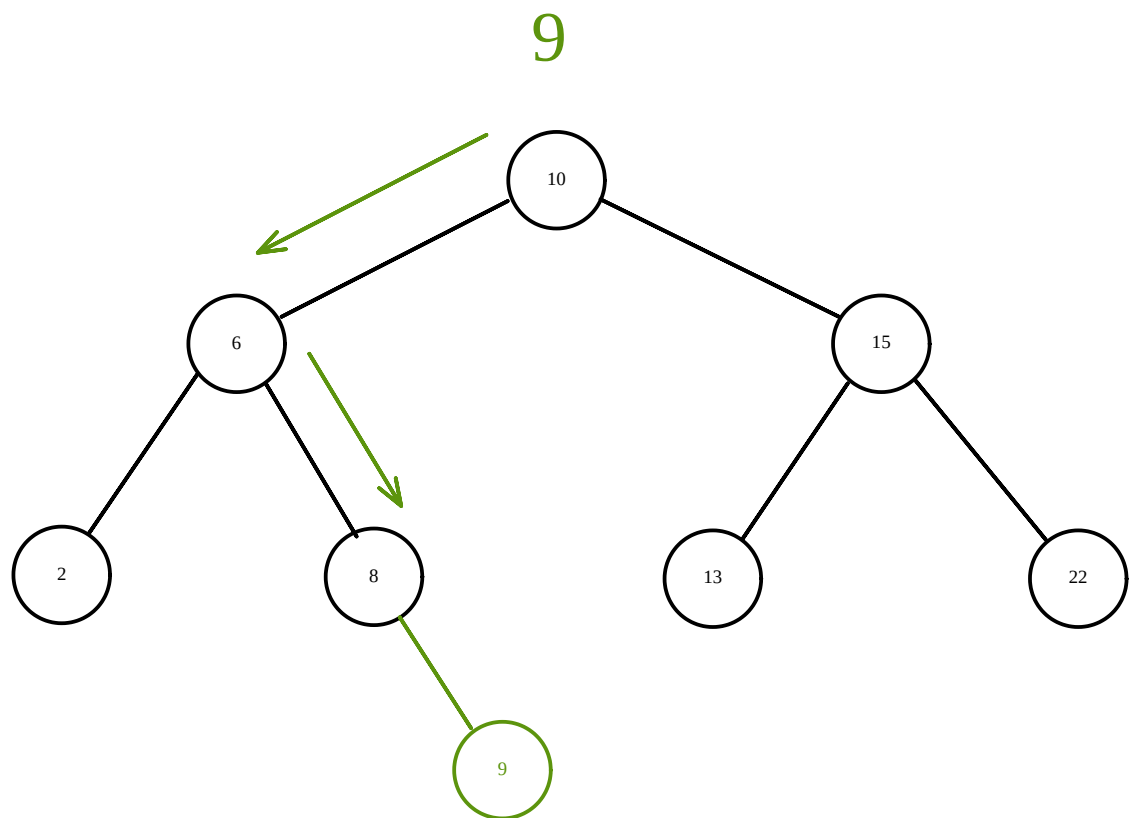
- Cada nó tem no máximo 2 filhos.
- E o filho a esquerda é menor que o pai,
- enquanto que o filho a direita é maior que o pai.
- Isso é bom por que ajuda na hora de pesquisar um item dentro dessa árvore

Árvore Binária



- **Lógica de Inserção**

- Começa na raiz, se for menor, vai pra esquerda, se for maior, vai pra direita.

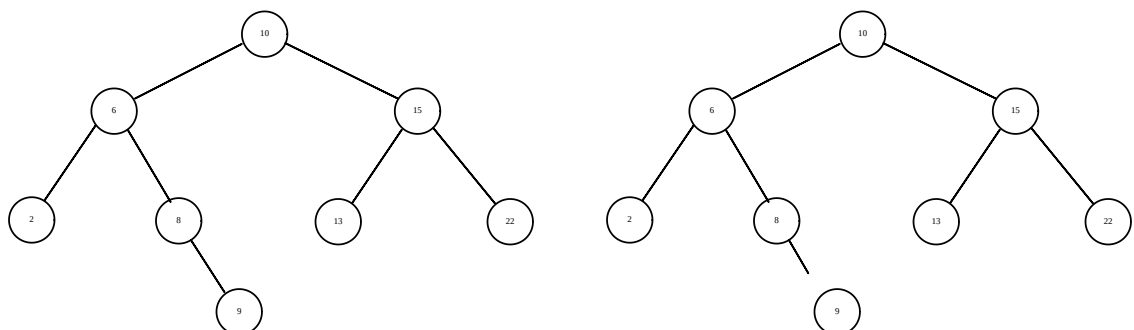


- **Lógica de Remoção**

1. Nó Folha

- a. Faça com que o seu pai aponte para null.

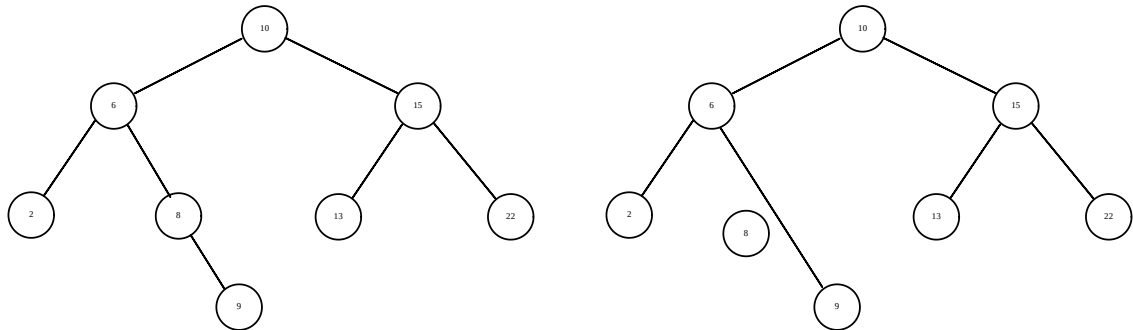
- i. 9



2. Nó com um filho.

a. Atualizar o pai desse nó, para que ele aponte para o seu neto diretamente.

i. 8



3. Nó com 2 filhos.

a. Ache o maior item da sub-árvore à esquerda.

i. Ou seja, seu predecessor.

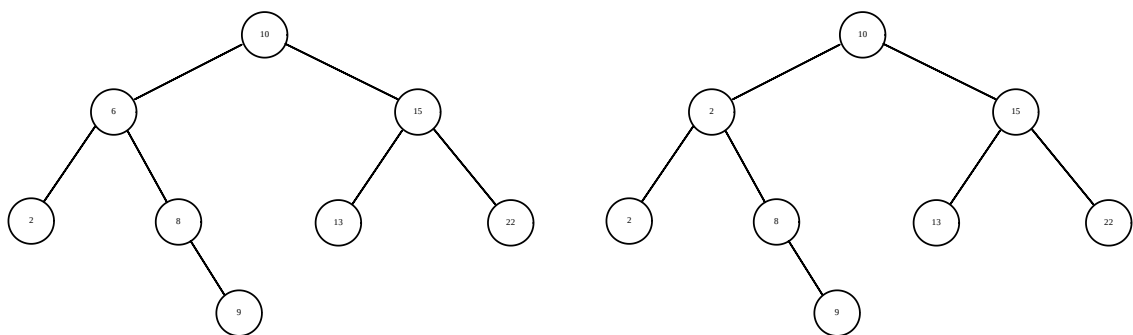
b. Ache o menor item da sub-árvore à direita.

a. Ou seja, seu sucessor.

c. Seja lá qual a escolha, o próximo passo é fazer com que o valor do item a ser excluído seja igual ao valor do seu predecessor ou sucessor.

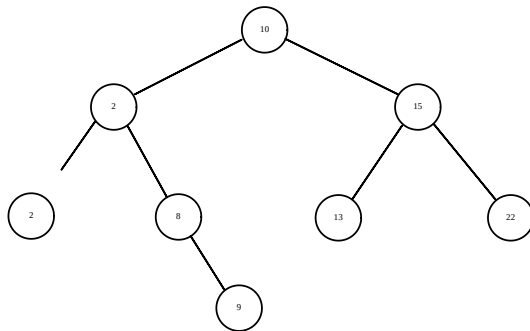
d. Depois, exclua o valor do predecessor ou sucessor e mantenha o novo valor onde está

a. 6



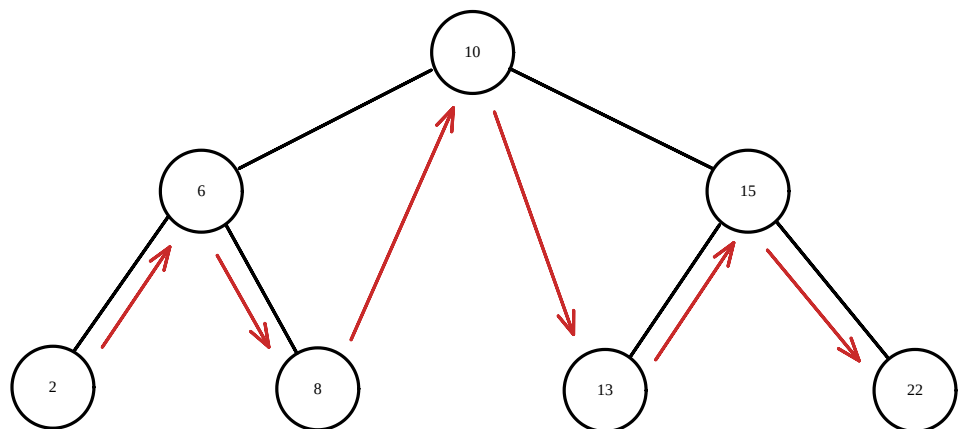
Árvore original

Achei o seu predecessor



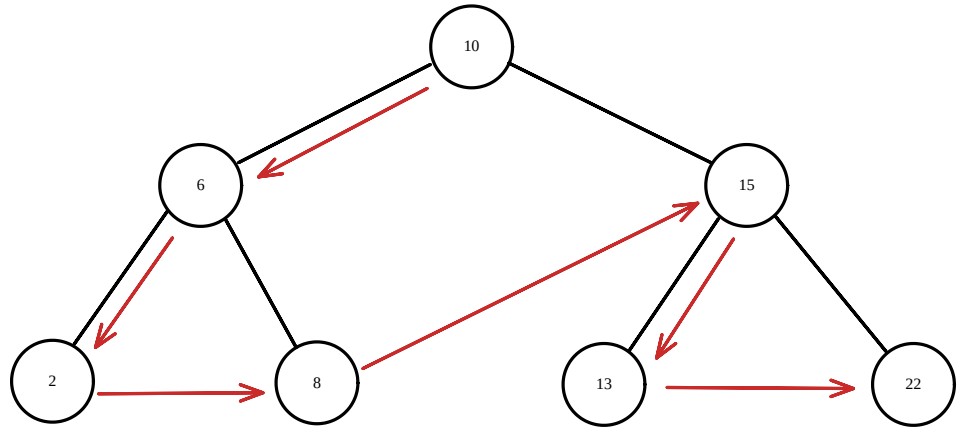
Nó excluído com sucesso.

- Atravessar uma árvore.
 - **Em ordem**
 - Esquerda, Raiz, Direita recursivamente.



2, 6, 8, 13, 15, 22

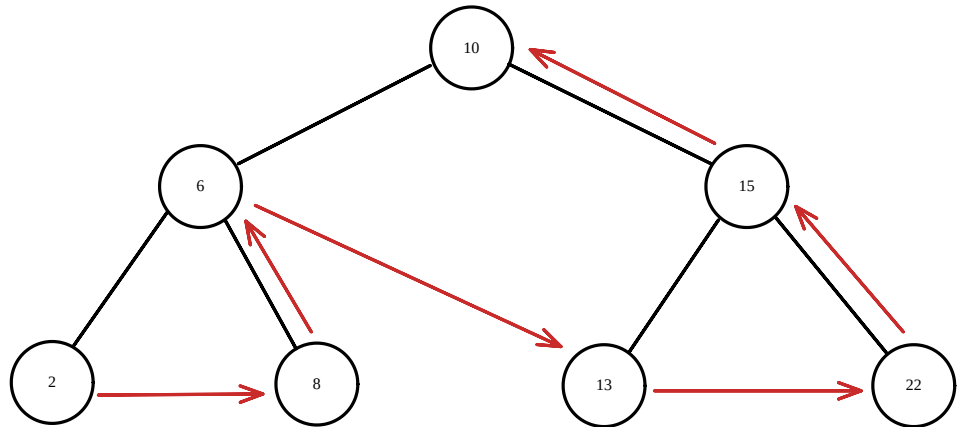
- Esse tipo de passagem vai ordenar todos os elementos de uma árvore.
- **Pré-Ordem**
 - Raiz, Esquerda, Direita



10, 6, 2, 8, 15, 13, 22

- **Pós-Ordem**

- Esquerda, Direita, Raiz.



2, 8, 6, 13, 22, 15, 10