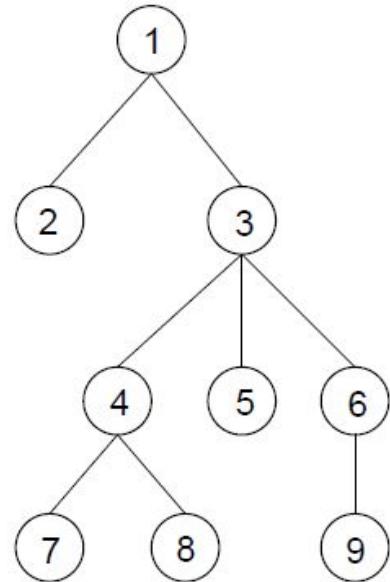


Estruturas de dados

Árvores binárias Busca e remoção

Definição de árvores

- **Raiz:** nó de origem da árvore.
- **Folhas:** nós que não têm filhos.
- **Grau de um nó:** é o número de subárvores (filhos) de um nó.
- **Nível de um nó:** número de nós no caminho da raiz até um nó.
- **Altura da árvore:** é o nível mais alto da árvore.
- **Sub-árvore:** todo nó contido na árvore é a raiz de uma sub-árvore.
- **Caminho:** é o caminho entre dois nós de um árvore.
- **Árvore cheia ou completa:** uma árvore que possui o número máximo de nós, isto é, todos os nós têm número máximo de filhos exceto as folhas, e todas as folhas estão na mesma altura.



Busca por elementos

- Buscar a(s) ocorrência(s) de um elemento em nós da árvore.

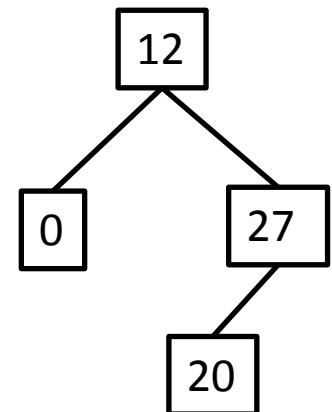
```

54 int busca (arv *a, int c) {
55     if (a == NULL) {
56         return 0; // árvore vazia
57     }
58     else {
59         if (c == a->valor) {
60             return 1;
61         }
62         else if (busca(a->esq, c) == 1) { // entra no ELSE IF se o retorno for 1
63             return 1;
64         }
65         else {
66             return busca(a->dir, c);
67         }
68     }
69 }

int main()
{
    //função para criar a árvore
    r = busca(a, 20);
}

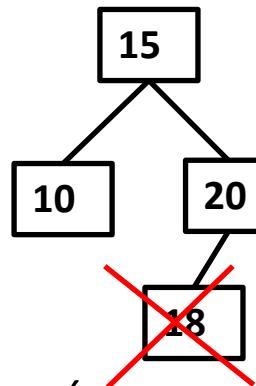
```

Realize o teste de mesa para a árvore deste exemplo.

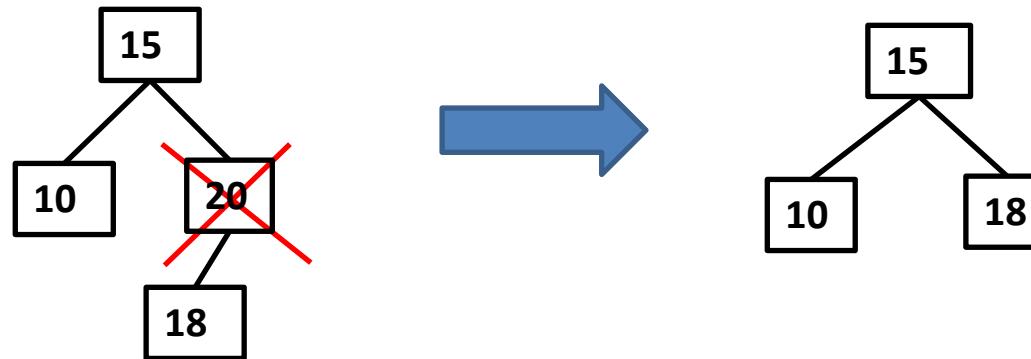


Remoção de elementos

- Existem 3 casos a considerar:
 - 1º caso: remoção de um nó folha: basta retirá-lo da árvore.



- 2º caso: Remoção de nó com apenas 1 filho: o filho do nó a ser retirado, ocupa seu lugar.

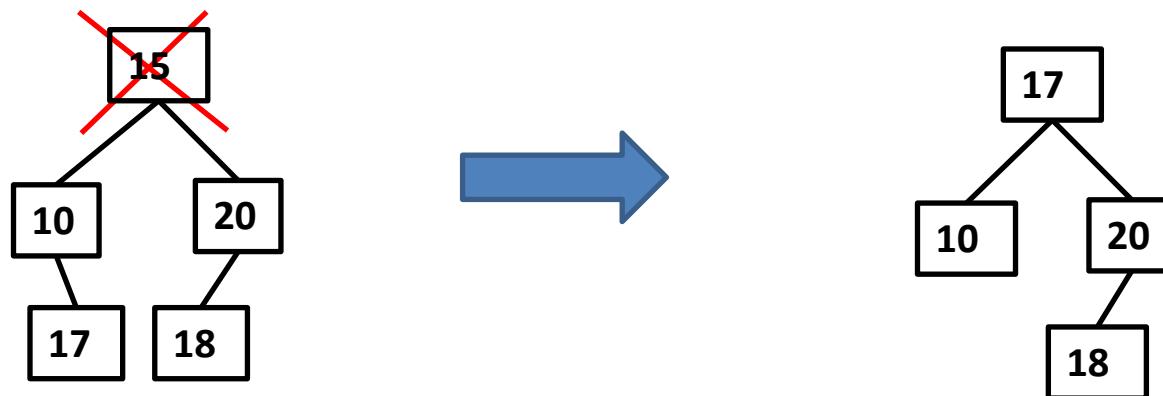


Remoção de elementos

- 3º caso: retirar nó com 2 filhos:
 - Substituir o elemento pelo nó mais à esquerda da subárvore da direita:

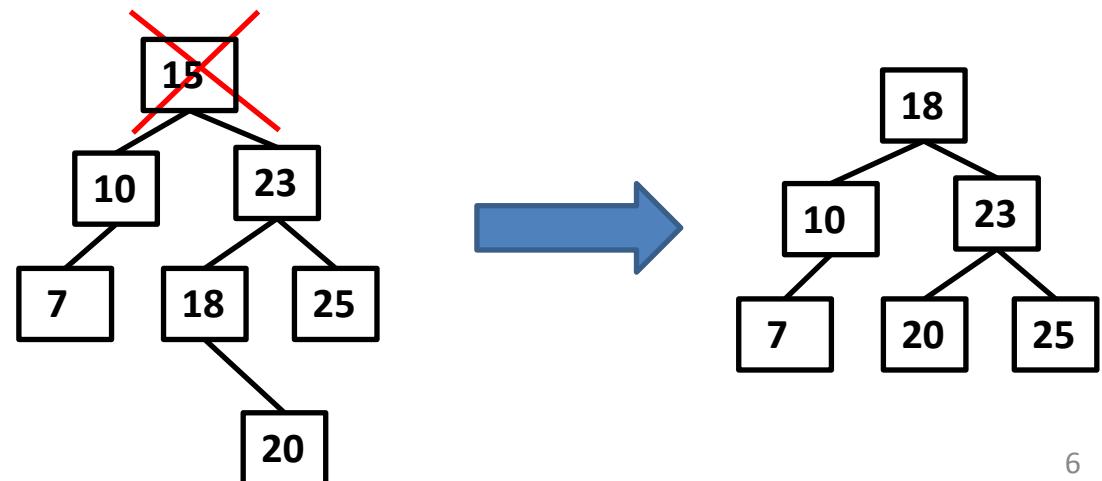
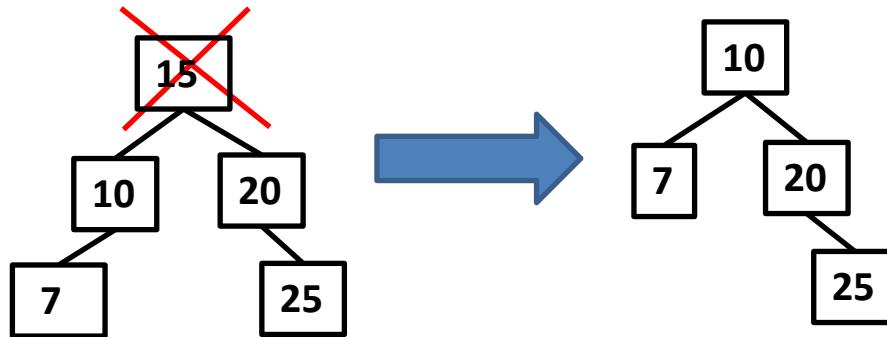


- Substituir o elemento pelo nó mais à direita da subárvore da esquerda:



Remoção de elementos

- Árvore binária de busca degenerada: São árvores com nós sem filhos à esquerda da subárvore da direita E sem filhos à direita na subárvore da esquerda: deve-se analisar a possibilidade de remoção.



Exercício

- A partir do código criado para os exercícios das aulas 8.1 e 8.2, desenvolva funções para:
 - Crie uma função para descobrir a altura da árvore.
 - Crie uma função para retirar elementos de uma árvore considerando as restrições estudadas.