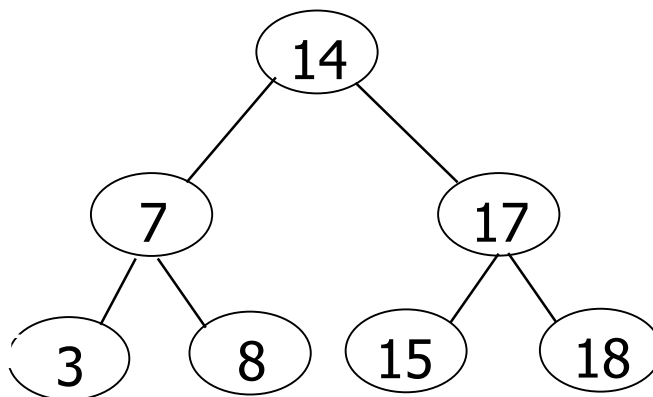


# Árvore Binária de Busca

# Definição

- Os nós pertencentes à sub-árvore esquerda possuem valores menores do que o valor associado ao nó-raiz  $r$
- Os nós pertencentes à sub-árvore direita possuem valores maiores do que o valor associado ao nó-raiz  $r$



# Operações - Inserção

- Passos do algoritmo de inserção
  - Procure um “local” para inserir o novo nó, começando a procura a partir do nó-raiz
  - Para cada nó-raiz de uma sub-árvore, compare;
    - se o novo nó possui um valor menor do que o valor no nó-raiz: vai para sub-árvore esquerda
    - se o valor é maior que o valor no nó-raiz: vai para sub-árvore direita
    - Se um ponteiro (filho esquerdo/direito de um nó-raiz) nulo é atingido, coloque o novo nó como sendo filho do nó-raiz

# Operações - Inserção

- **Exemplo:**
  - Inserção do conjunto de números, na sequência:
    - 17, 99, 13, 1, 3, 100

# Operações - Busca

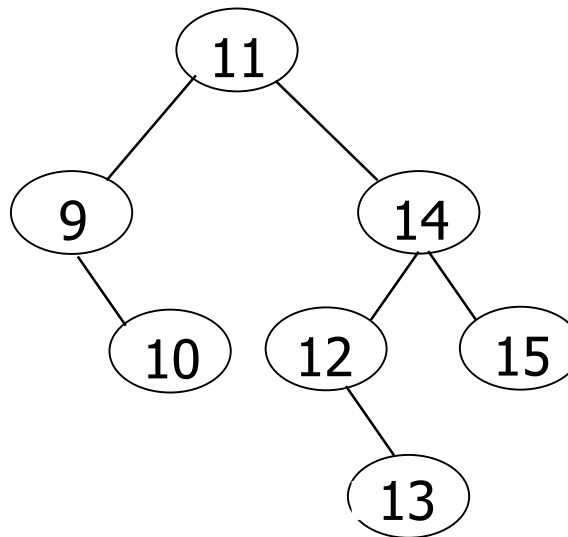
- Passos do algoritmo de inserção
  - Comece a busca a partir do nó-raiz
  - Para cada nó-raiz de uma sub-árvore compare:
    - se o valor procurado é menor que o valor no nó-raiz:  
continua pela sub-árvore esquerda
    - se o valor é maior que o valor no nó-raiz: sub-árvore  
direita
    - Caso o nó contendo a chave pesquisada seja encontrado,  
retorne true e o nó pesquisado, caso contrário retorne false

# Operações - Remoção

- Casos a serem considerados no algoritmo de remoção de nós de uma ABB:
  - Caso 1: o nó é folha:
    - O nó pode ser retirado sem problema
  - Caso 2: o nó possui uma sub-árvore (esq/dir):
    - O nó-raiz da sub-árvore (esq/dir) “ocupa” o lugar do nó retirado
  - Caso 3: o nó possui duas sub-árvores:
    - O nó contendo o menor valor da sub-árvore direita pode “ocupar” o lugar
    - Ou o maior valor da sub-árvore esquerda pode “ocupar” o lugar

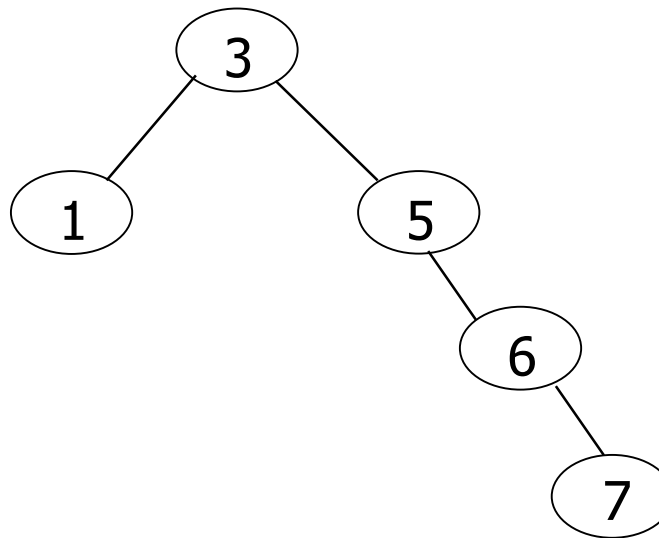
# Operações - Remoção

- Caso 1
  - Remoção do 15



# Operações - Remoção

- Caso 2
  - Remoção do 5





# Operações - Remoção

- Caso 3
  - Remoção do 11

