Algoritmos e Estrutura de Dados

Bruno Feres de Souza
bferes@gmail.com

Universidade Federal do Maranhão
Bacharelado em Ciência e Tecnologia

1º semestre de 2016

Na aula passada...

Árvore Binária de Busca

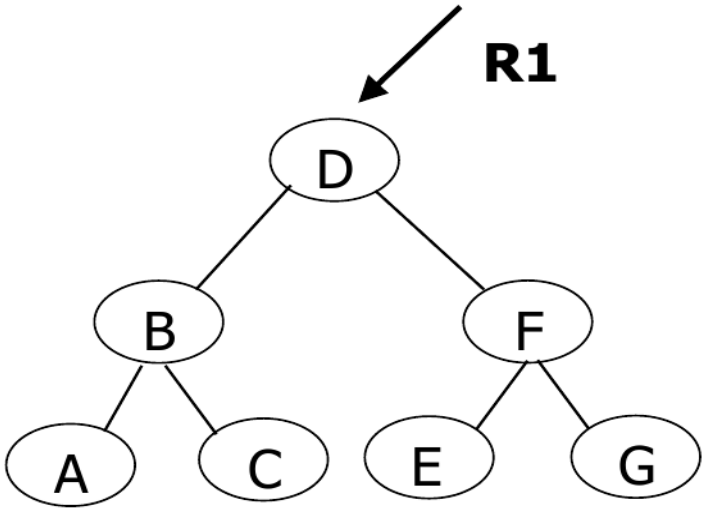
- Uma **Árvore Binária de Busca (ABB)** possui as seguintes propriedades:
  - É uma árvore binária
  - Seja $S=\{s_1, s_2,...,s_n\}$ o conjuntos de chaves (informações) dos nós da árvore T
    - Esse conjunto satifaz $s_1 < s_2 < \ldots < s_n$
    - A cada nó v_j de T está associada uma chave distinta s_j de S, que pode ser consultado por $r(v_j)=s_j$
  - Dado um nó v de T
    - Se v_i pertence à sub-árvore à esquerda de v, então $r(v_i) < r(v)$
    - Se v_i pertence à sub-árvore à direita de v, então $r(v_i) > r(v)$

Árvore Binária de Busca

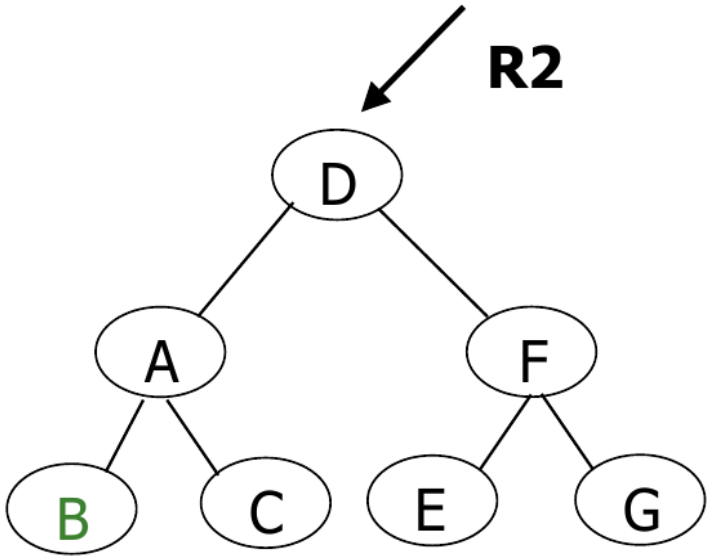
- Em outras palavras, em uma **ABB**, tem-se que
  - Para qualquer nó v, os nós de sua sub-árvore à esquerda possuem valores menores do que o valor associado a v
  - Para qualquer nó v, os nós de sua sub-árvore à direita possuem valores maiores do que o valor associado a v
  - As sub-árvores à esquerda e à direita são ABB

Árvore Binária de Busca

- Exemplos



é ABB



não é ABB

Árvore Binária de Busca

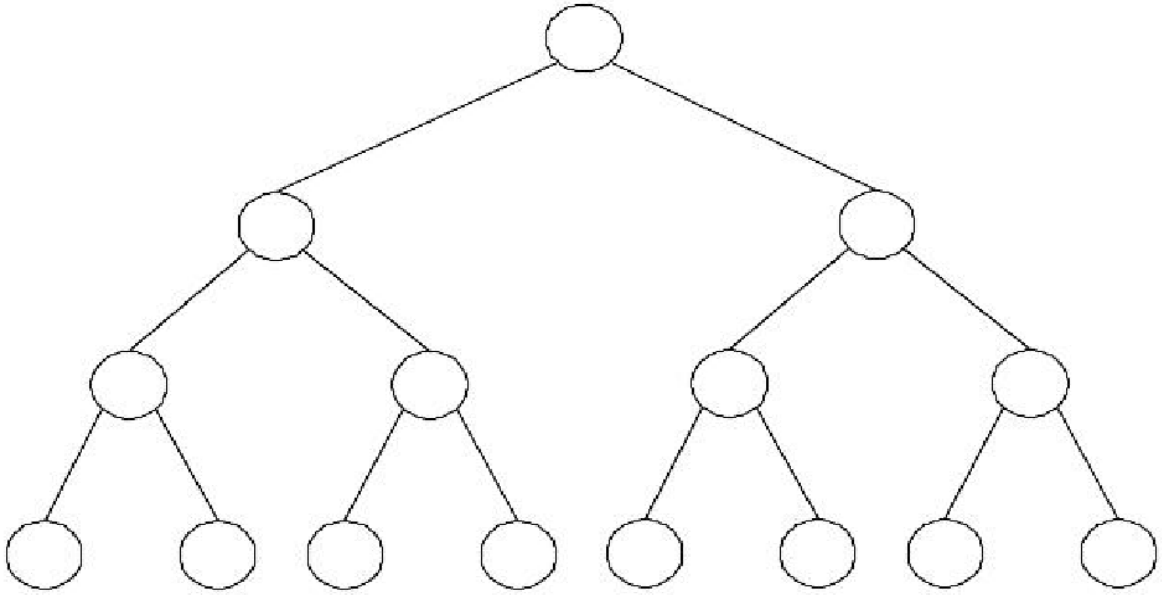
- Por que utilizar ABB?
  - Imagine um sistema de votação por telefone
    - Cada número só pode votar uma vez
    - O sistema deve armazenar todos os números que já votaram
    - A cada ligação deve-se verificar se o número já votou
    - A votação deve ter resultado em tempo real

Árvore Binária de Busca

- Por que utilizar ABB?
  - Implementação do sistema por ABB
    - Cada número é armazenado em um nó da ABB
    - Suponha que em um determinado momento, a ABB tenha milhões de telefones armazenados
    - Surge uma nova ligação e é preciso saber se o número está ou não na árvore (já votou ou não)

Árvore Binária de Busca

- Por que utilizar ABB?
  - Implementação do sistema por ABB
    - Considere, neste caso, que esta ABB é uma Árvore Binária Cheia



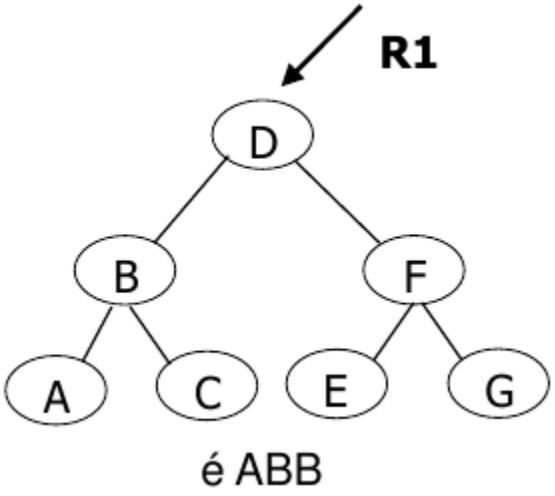
0	1
1	3
2	7
...	...
10	2047
16	131071

Árvore Binária de Busca

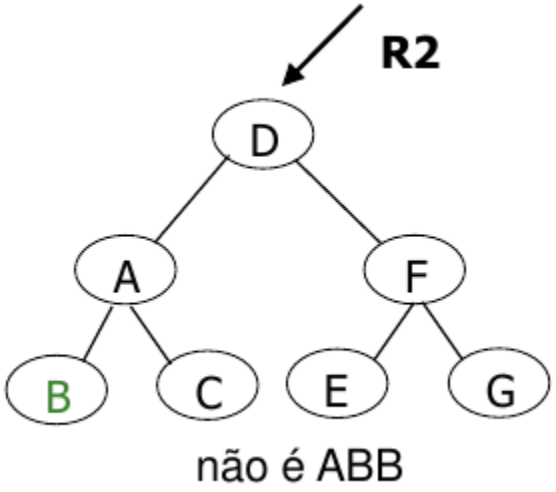
- Por que utilizar ABB?
  - Implementação do sistema por ABB
    - Considere, neste caso, que esta ABB é uma Árvore Binária Cheia
      - » Para buscar um elemento na árvore, percorre-se os nós da raiz atá as folhas, sem passar por mais de um nó em um mesmo nível
      - » Portanto, no pior caso, a busca passa por tantos nós quanto for a altura da árvore

Árvore Binária de Busca

- Por que utilizar ABB?
  - Implementação do sistema por ABB
    - Exemplo: buscar pelo elemento **E** nas árvores abaixo



3 consultas



6 consultas

Árvore Binária de Busca

- Por que utilizar ABB?
  - Buscas muito rápidas!

Árvore Binária de Busca

- Operação básicas em uma ABB
  - Buscar um elemento
  - Inserir um elemento
  - Remover um elemento

OBS: as operações devem considerar a ordenação dos elementos na ABB. Por exemplo, na inserção, deve-se procurar pelo lugar certo para Inserir o elemento.

Árvore Binária de Busca

- Operação de Busca
  - Comparando a “chave” com a informação no nó raiz, quatro casos pode ocorrer:
  - A árvore é vazia → a chave não está na árvore → Fim
  - A chave buscada está na raiz → elemento encontrado → Fim
  - A chave é menor do que a informação da raiz → fazer busca na sub-árvore da esquerda
  - A chave é maior do que a informação da raiz → fazer busca na sub-árvore da direita

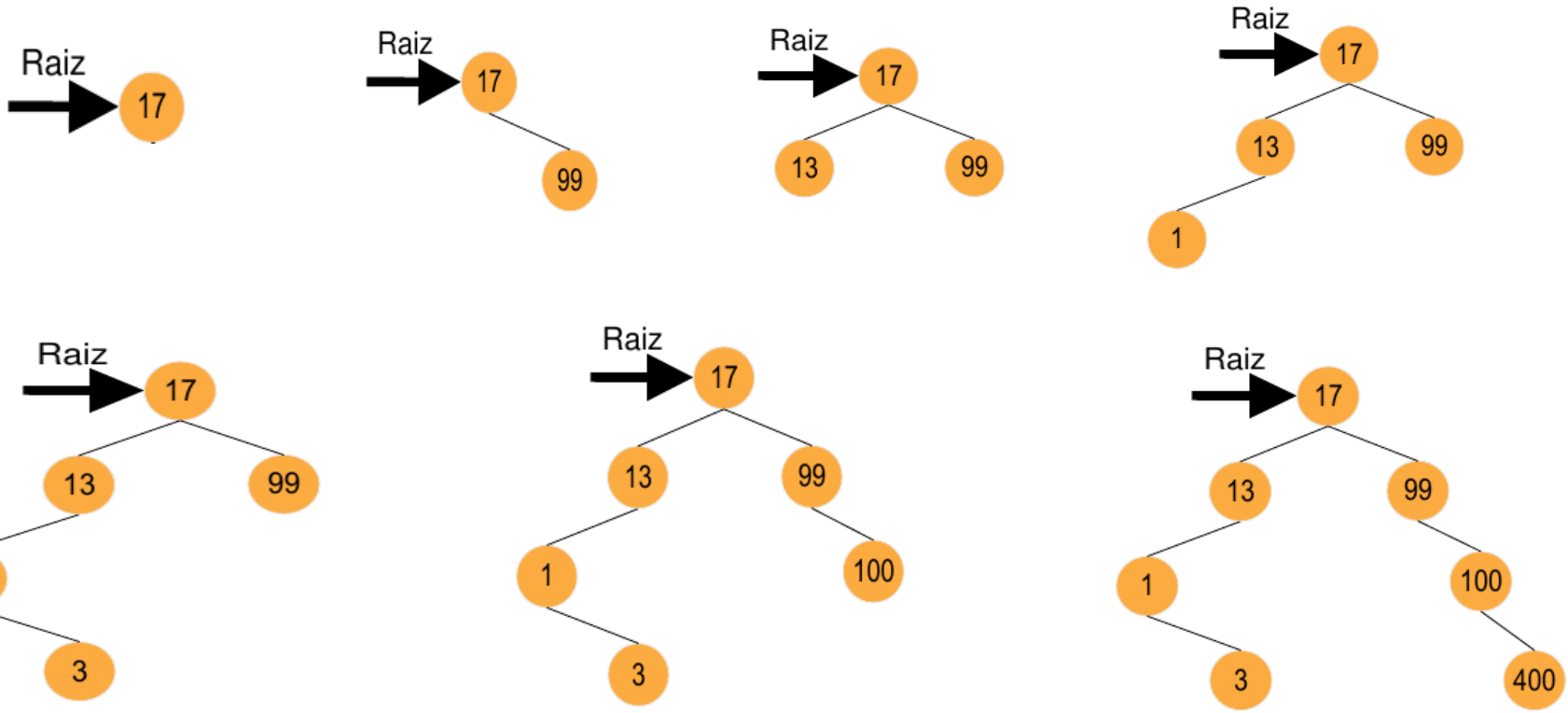
OBS: os mesmo casos acontecem quando da busca nas sub-árvores

Árvore Binária de Busca

- Operação de Inserção
  - Estratégia geral: sempre inserir elementos como nós folhas, procurando a posição correta.
  - Comparando a “chave” com a informação no nó raiz, quatro casos pode ocorrer:
  - A árvore é vazia → insere o elemento, que passará a ser a raiz → Fim
  - A chave buscada está na raiz → o elemento já está na árvore → Fim
  - A chave é menor do que a informação da raiz → insere na sub-árvore da esquerda
  - A chave é maior do que a informação da raiz → insere na sub-árvore da direita

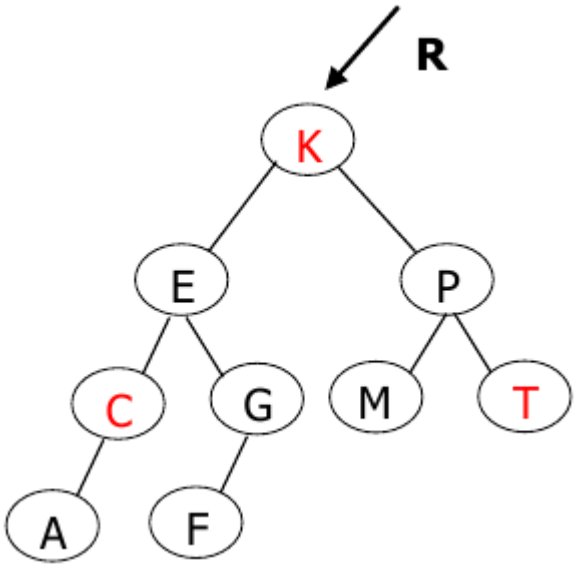
Árvore Binária de Busca

- Operação de Inserção
  - Exemplo: inserir os elementos {17,99,13,1,3,100,400} em uma ABB vazia.



Árvore Binária de Busca

- Operação de Remoção
  - Exemplo: para a árvore abaixo, remova os elementos T, C e K, nesta ordem. Perceba que cada caso é diferente.



Árvore Binária de Busca

- Operação de Remoção

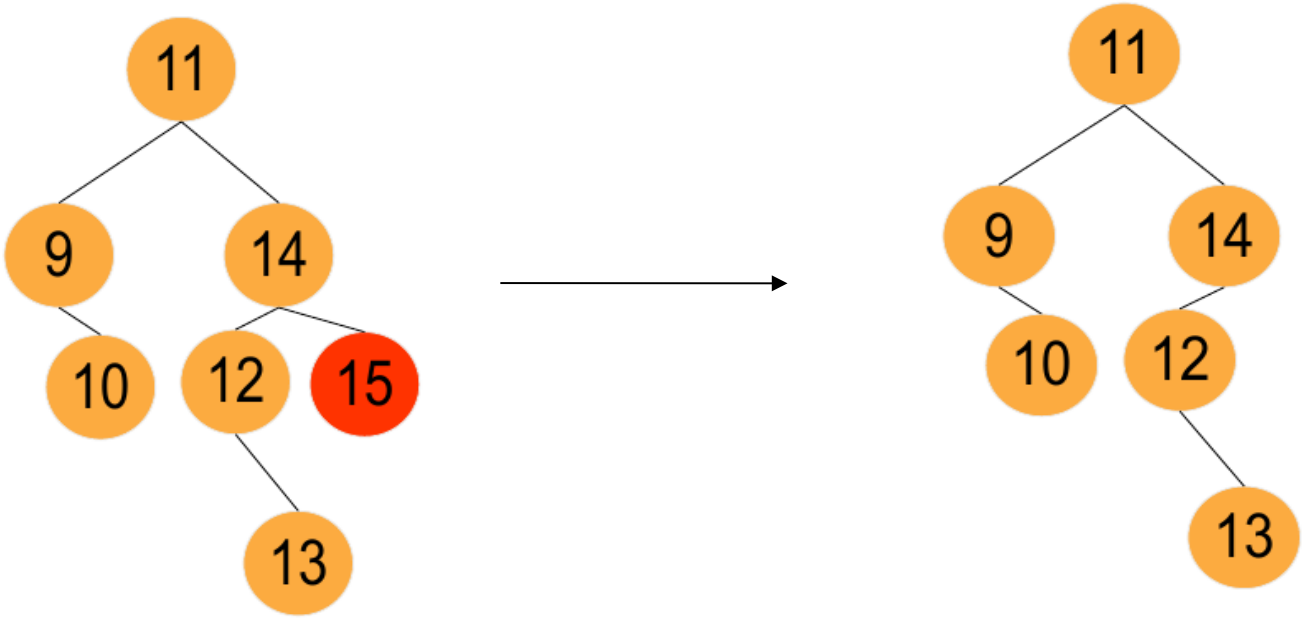
  - Caso 1 (remover T): o nó a ser removido não tem filho
  - Remove-se o nó
  - Quem apontava pra ele recebe None

  - Caso 2 (remover C): o nó a ser removido tem um único filho
  - “Puxa-se” o filho para o lugar do pai
  - Remove-se o nó

  - Caso 3 (remover K): o nó a ser removido tem 2 filhos
  - Acha-se o maior valor de informação da sub-árvore à esquerda (ou o menor da direita)
  - O nó a ser removido recebe essa informação
  - Remove-se o nó com o maior valor de informação da subárvore à esquerda (ou o menor da direita)

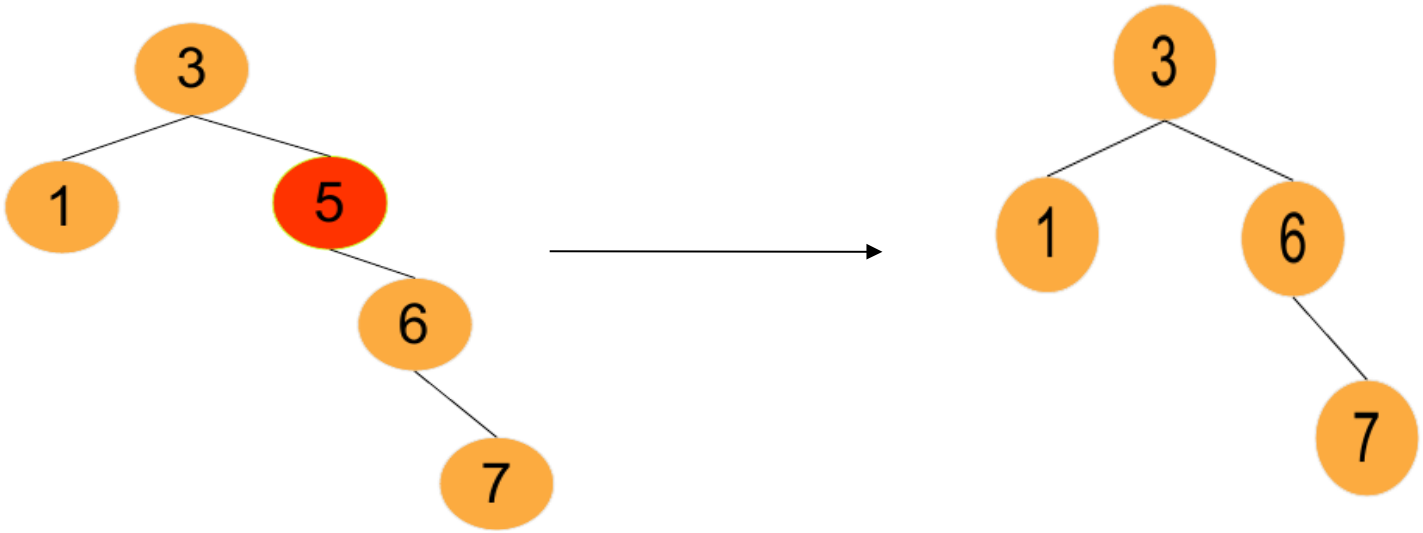
Árvore Binária de Busca

- Operação de Remoção
  - Caso 1: o nó a ser removido não tem filho
  - Remove-se o nó
  - Quem apontava pra ele recebe NULL



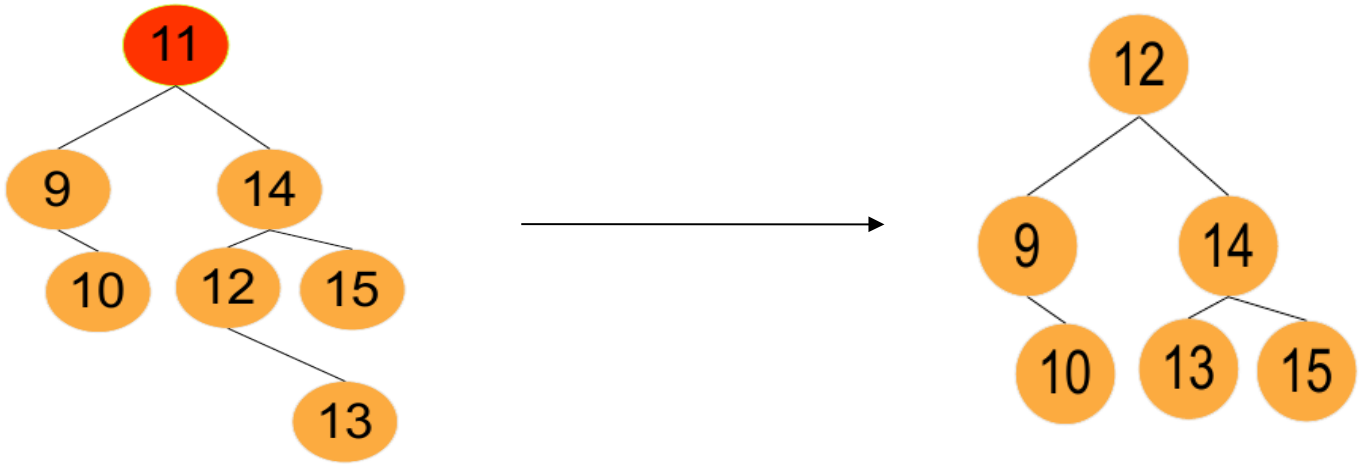
Árvore Binária de Busca

- Operação de Remoção
  - Caso 2 (remover C): o nó a ser removido tem um único filho
  - “Puxa-se” o filho para o lugar do pai
  - Remove-se o nó



Árvore Binária de Busca

- Operação de Remoção
  - Caso 3 (remover K): o nó a ser removido tem 2 filhos
  - Acha-se o maior valor de informação da sub-árvore à esquerda (ou o menor da direita)
  - O nó a ser removido recebe essa informação
  - Remove-se o nó com o maior valor de informação da subárvore à esquerda (ou o menor da direita)



Dúvidas?