

Árvores Binárias de Busca

Estruturas de Dados I

Departamento de Computação

Universidade Federal de São Carlos (UFSCar)

Sumário

- 1 Conceitos Introdutórios
- 2 Inserção em Árvores Binárias de Busca
- 3 Pesquisa em Árvores Binárias de Busca
- 4 Remoção em Árvores Binárias de Busca
- 5 Conceitos Adicionais

Definições

- Uma **Árvore Binária de Busca (ABB)** possui as seguintes propriedades
 - É uma Árvore Binária (AB)
 - Seja $S = \{s_1, \dots, s_n\}$ o conjunto de chaves dos nós da árvore T
 - Esse conjunto satisfaz $s_1 < \dots < s_n$
 - A cada nó $v_j \in T$ está associada uma chave distinta $s_j \in S$, que pode ser consultada por $r(v_j) = s_j$
 - Dado um nó v de T
 - Se v_i pertence a sub-árvore esquerda de v , então $r(v_i) < r(v)$
 - Se v_i pertence a sub-árvore direita de v , então $r(v_i) > r(v)$

Definições

- Em outras palavras
 - 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

Definições

- Um **percurso em-ordem** em uma ABB resulta na sequência de valores em **ordem crescente**
- Se **invertêssemos as propriedades** descritas na definição anterior, de maneira que a sub-árvore esquerda de um nó contivesse valores maiores e a sub-árvore direita valores menores, o percurso em-ordem resultaria nos valores em **ordem decrescente**
- Uma **ABB** criada a partir de um conjunto de valores **não é única**: o resultado depende da sequência de inserção dos dados

Definições

- A grande utilidade da árvore binária de busca é armazenar dados contra os quais outros dados são frequentemente verificados (busca!)
- Uma árvore de binária de busca é dinâmica e pode sofrer alterações (inserções e remoções de nós) após ter sido criada

Operações em ABB's

- Inserção
- Pesquisa
- Remoção

Sumário

- 1 Conceitos Introdutórios
- 2 Inserção em Árvores Binárias de Busca
- 3 Pesquisa em Árvores Binárias de Busca
- 4 Remoção em Árvores Binárias de Busca
- 5 Conceitos Adicionais

Inserção (operações em ABB's)

- 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), ou 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

Inserção

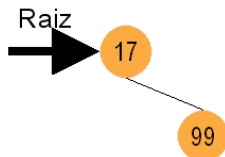
- Para entender o algoritmo considere a inserção do conjunto de números, na sequência

17,99,13,1,3,100,400

- No início a ABB está vazia!

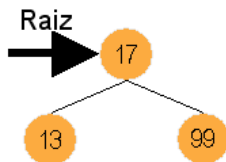
Inserção

- O número 17 será inserido tornando-se o nó raiz
- A inserção do 99 inicia-se na raiz. Compara-se 99 com 17
- Como $99 > 17$, 99 deve ser colocado na sub-árvore direita do nó contendo 17 (subárvore direita, inicialmente, nula)



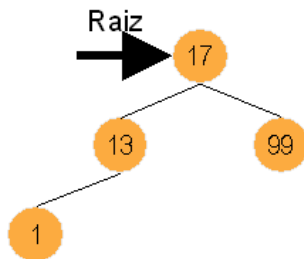
Inserção

- A inserção do 13 inicia-se na raiz
- Compara-se 13 com 17.
Como $13 < 17$, 13 deve ser colocado na sub-árvore esquerda do nó contendo 17
- Já que o nó 17 não possui descendente esquerdo, 13 é inserido na árvore nessa posição



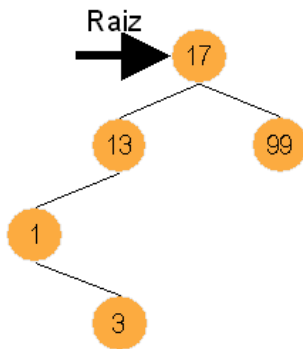
Inserção

- Repete-se o procedimento para inserir o valor 1
- $1 < 17$, então será inserido na sub-árvore esquerda
- Chegando nela, encontra-se o nó 13, $1 < 13$ então ele será inserido na sub-árvore esquerda de 13



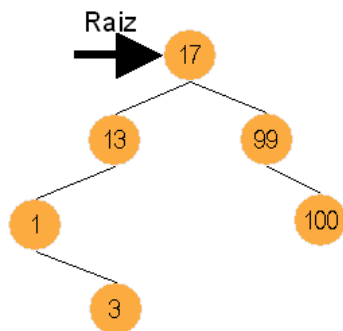
Inserção

- Repete-se o procedimento para inserir o elemento 3
 - $3 < 17$
 - $3 < 13$
 - $3 > 1$



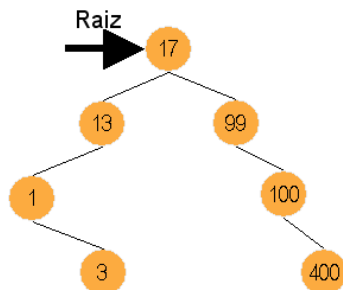
Inserção

- Repete-se o procedimento para inserir o elemento 100
 - $100 > 17$
 - $100 > 99$



Inserção

- Repete-se o procedimento para inserir o elemento 400
 - $400 > 17$
 - $400 > 99$
 - $400 > 100$



Sumário

- 1 Conceitos Introdutórios
- 2 Inserção em Árvores Binárias de Busca
- 3 Pesquisa em Árvores Binárias de Busca**
- 4 Remoção em Árvores Binárias de Busca
- 5 Conceitos Adicionais

Pesquisa (operações em ABB's)

- Passos do algoritmo de busca
 - 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), ou 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**

Sumário

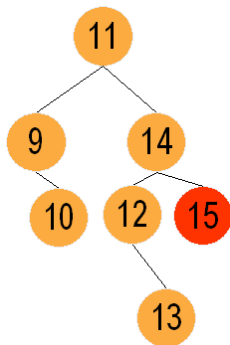
- 1 Conceitos Introdutórios
- 2 Inserção em Árvores Binárias de Busca
- 3 Pesquisa em Árvores Binárias de Busca
- 4 Remoção em Árvores Binárias de Busca**
- 5 Conceitos Adicionais

Remoção (operações em ABB's)

- 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

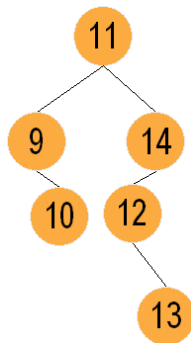
Remoção - Caso 1

- Caso o valor a ser removido seja o 15
- Pode ser removido sem problema, não requer ajustes posteriores



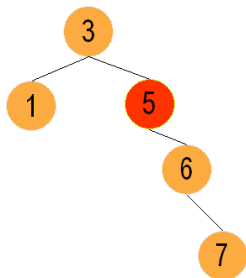
Remoção - Caso 1

- Os nós com os valores 10 e 13 também podem ser removidos



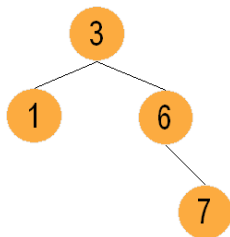
Remoção - Caso 2

- Removendo-se o nó com o valor 5
- Como ele possui somente a sub-árvore direita, o nó contendo o valor 6 pode “ocupar” o lugar do nó removido



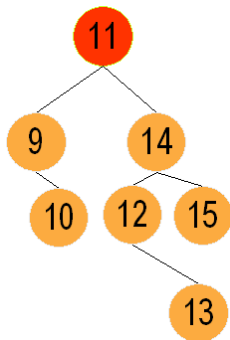
Remoção - Caso 2

- Esse segundo caso é análogo, caso existisse um nó com somente uma sub-árvore esquerda



Remoção - Caso 3

- Eliminando-se o nó de chave 11
- Neste caso, existem 2 opções
 - O nó com chave 10 pode “ocupar” o lugar do nó-raiz, ou
 - O nó com chave 12 pode “ocupar” o lugar do nó-raiz



Sumário

- 1 Conceitos Introdutórios
- 2 Inserção em Árvores Binárias de Busca
- 3 Pesquisa em Árvores Binárias de Busca
- 4 Remoção em Árvores Binárias de Busca
- 5 Conceitos Adicionais**

Complexidade da busca em ABB

- Pior caso
 - Número de passos é determinado pela altura da árvore
 - Árvore degenerada possui altura igual a n
- Altura da árvore depende da sequência de inserção das chaves
 - O que acontece se uma sequência ordenada de chaves é inserida
- Busca eficiente se árvore razoavelmente balanceada

Árvores Binárias de Busca

- ABB “aleatória”
 - Nós externos: descendentes dos nós folha (não estão, de fato, na árvore)
 - Uma árvore A com n nós possui $n + 1$ nós externos
 - Uma inserção em A é considerada “aleatória” se ela tem probabilidade igual de acontecer em qualquer um dos $n + 1$ nós externos
 - Uma ABB aleatória com n nós é uma árvore resultante de n inserções aleatórias sucessivas em uma árvore inicialmente vazia

Árvores Binárias de Busca

- É possível demonstrar que para uma ABB “aleatória” o número esperado de comparações para recuperar um registro qualquer é cerca de $1,39 * \log_2(n)$
 - 39% pior do que o custo do acesso em uma árvore balanceada
- Pode ser necessário garantir um melhor balanceamento da ABB para melhor desempenho na busca

Árvores Binárias de Busca

- A complexidade das operações de inserção e remoção também dependem da eficiência da busca
- O tempo necessário para realizar essas operações depende principalmente do tempo necessário para encontrar a posição do nó a ser inserido/removido
- A remoção ainda pode requerer encontrar o nó máximo da sub-árvore esquerda (*troca_max_esq(...)*), mas o número de operações realizadas é sempre menor ou igual do que a altura da árvore

Exercícios

- Escreva uma função que verifique se uma árvore binária está perfeitamente balanceada
 - O número de nós de suas sub-árvores esquerda e direita difere em, no máximo, 1

Leitura Complementar

1. *Árvore Binária de Busca*

http://pt.wikipedia.org/wiki/Árvore_binária_de_busca

2. *Binary trees*

<http://cslibrary.stanford.edu/110/>

3. *Binary search tree*

<http://xlinux.nist.gov/dads//HTML/binarySearchTree.html>

Créditos: aula baseada nos tópicos de aula e slides criados pelo prof. Fernando Vieira Paulovich, publicamente disponíveis em: [http://wiki.icmc.usp.br/index.php/Scc-202\(paulovich\)](http://wiki.icmc.usp.br/index.php/Scc-202(paulovich))