



UNIVERSIDADE FEDERAL DO CEARÁ
Campus de Quixadá
Prof. Arthur Araruna
QXD0115- Estrutura de Dados Avançada

LE2
2025.1

Nome: _____ Matrícula: _____

Observação

Para os exercícios a seguir, quando alguma questão solicitar um algoritmo que recebe ou retorna uma árvore, considere que essa árvore será sempre representada por seu nó raiz.

1 Exercícios de Fixação

- Iniciando com uma árvore binária de busca vazia, para cada item, como fica essa árvore após a inserção dos valores na ordem apresentada?
 - $\langle 30, 25, 20, 15, 10, 5, 0 \rangle$
 - $\langle 30, 25, 20, 0, 5, 10, 15 \rangle$
 - $\langle 0, 5, 30, 15, 20, 25, 10 \rangle$
- POSCOMP 2017 (adaptada)**

O percurso em pré-ordem *fixado à esquerda* de uma Árvore Binária de Busca é $\langle 44, 30, 12, 26, 36, 33, 92, 64, 46, 98 \rangle$. Um percurso em pré-ordem *fixado à direita* da mesma árvore é:

OBS: A ordem de fixação diz respeito à ordem entre as sub-árvores. Se a sub-árvore esquerda vem antes da direita, o percurso é fixado à esquerda.

 - $\langle 26, 12, 33, 36, 30, 46, 64, 98, 92, 44 \rangle$
 - $\langle 44, 92, 98, 64, 46, 30, 36, 33, 12, 26 \rangle$
 - $\langle 12, 26, 30, 33, 36, 44, 46, 64, 92, 98 \rangle$
 - $\langle 98, 46, 64, 92, 33, 36, 26, 12, 30, 44 \rangle$
 - $\langle 98, 92, 64, 46, 44, 36, 33, 30, 26, 12 \rangle$
- Suponha que possuímos números entre 1 e 1000 em uma *árvore binária de busca* e realizamos uma busca pelo valor 363. Diga quais das sequências abaixo não podem ter sido a sequência de valores pelos quais passamos na árvore durante essa busca e por que essas sequências seriam impossíveis de acontecer.
 - $\langle 2, 399, 387, 219, 266, 382, 381, 278, 363 \rangle$
 - $\langle 2, 252, 401, 398, 330, 344, 397, 363 \rangle$
 - $\langle 935, 278, 347, 621, 299, 392, 358, 363 \rangle$
 - $\langle 924, 220, 911, 244, 898, 258, 362, 363 \rangle$
 - $\langle 925, 202, 911, 240, 912, 245, 363 \rangle$
- A ABB ilustrada na Figura 1 na página seguinte foi construída inserindo uma sequência de elementos a partir de uma árvore vazia. Quais das sequências a seguir não produziram a árvore ilustrada?
 - $\langle 5, 3, 4, 9, 12, 7, 8, 6, 20 \rangle$
 - $\langle 5, 9, 3, 7, 6, 4, 8, 12, 20 \rangle$
 - $\langle 5, 9, 7, 8, 6, 12, 20, 3, 4 \rangle$
 - $\langle 5, 9, 7, 3, 8, 12, 6, 4, 20 \rangle$
 - $\langle 5, 9, 3, 6, 7, 8, 4, 12, 20 \rangle$

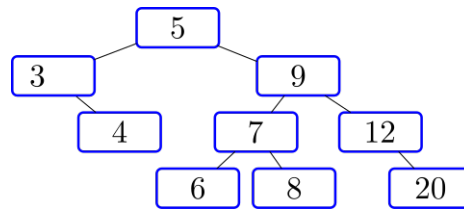


Figura 1:

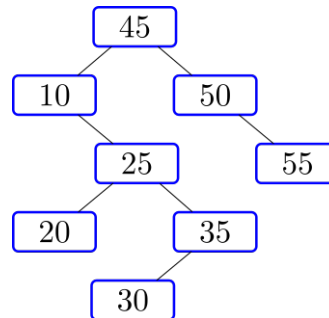


Figura 2:

5. Considere a árvore binária de busca na Figura 2
- Liste todas as possíveis ordens de inserção dos elementos que podem ter produzido essa árvore.
 - Faça um desenho de como ficará essa árvore após a inserção dos elementos 5, 40, 32, 90, 60, 65, nessa ordem.
 - Desenhe a árvore resultante da remoção dos elementos 25 e 55 da árvore do item b.
6. Dado o vetor ordenado $V = [5, 10, 15, 20, 25, 30, 35, 40]$ e uma ABB T inicialmente vazia, sugira uma sequência de inserção dos elementos de V em T de forma que:
- T resulte em uma árvore de altura máxima.
 - T resulte em uma árvore de altura mínima.

7. **POSCOMP 2014.**

Sobre árvores binárias, considere as afirmativas a seguir.

- Qualquer nó de uma árvore binária é raiz de, no máximo, outras duas subárvores comumente denominadas subárvore direita e subárvore esquerda.
- Uma dada árvore binária A armazena números inteiros e nela foram inseridos 936 valores não repetidos. Para determinar se um número x está entre os elementos dessa árvore, tal número será comparado, no máximo, com 10 números contidos na árvore A .
- Uma dada árvore binária de busca A armazena números inteiros e nela foram inseridos 936 valores não repetidos. Para determinar se um número x está entre os elementos dessa árvore, serão feitas, no máximo, 10 comparações.
- Uma dada árvore binária de busca A armazena números inteiros e nela foram inseridos 936 valores não repetidos. Supondo que r seja o nó raiz da árvore A e que sua subárvore esquerda contenha 460 elementos e sua subárvore direita possua 475 elementos. Para determinar se um número x pertence a essa árvore, serão feitas, no máximo, 476 comparações.

Assinale a alternativa correta.

- Somente as afirmativas I e II são corretas.
- Somente as afirmativas I e IV são corretas.
- Somente as afirmativas III e IV são corretas.
- Somente as afirmativas I, II e III são corretas.

E. Somente as afirmativas II, III e IV são corretas.

8. **ENADE 2014. (adaptada)**

A Figura 3 a seguir apresenta uma árvore binária de busca, que mantém a seguinte propriedade fundamental: o valor associado à raiz é sempre menor do que o valor de todos os nós da subárvore à direita e sempre menor do que o valor de todos os nós da subárvore à esquerda.

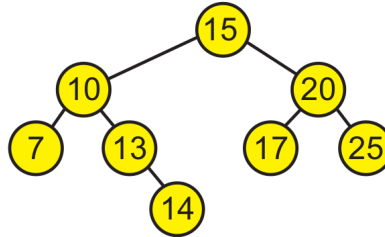


Figura 3:

Em relação à árvore apresentada na figura, avalie as afirmações a seguir:

- I. A árvore possui a vantagem de realizar a busca de elementos de forma eficiente, como a busca binária em um vetor.
- II. A árvore está desbalanceada, pois a subárvore da esquerda possui um número de nós maior do que a subárvore da direita.
- III. Quando a árvore é percorrida utilizando o método de percurso pós-ordem, os valores são encontrados em ordem decrescente.
- IV. O número de comparações realizadas em função do número n de elementos na árvore em uma busca binária realizada com sucesso é $O(\lg n)$.

É correto apenas o que se afirma em:

- A. I e III.
- B. I e IV.
- C. II e III.
- D. I, II e IV.
- E. II, III e IV.

9. **ENADE 2008. (adaptada)**

Um programador propôs um algoritmo não-recursivo para o percurso em pré-ordem de uma árvore binária com as seguintes características.

- Cada nó da árvore binária é representado por um registro com três campos: chave, que armazena seu identificador; *esq* e *dir*, ponteiros para os filhos esquerdo e direito, respectivamente.
- O algoritmo deve ser invocado inicialmente tomando o ponteiro para o nó raiz da árvore binária como argumento.
- O algoritmo utiliza `EMPILHA()` e `DESEMPILHA()` como funções auxiliares de empilhamento e desempilhamento de nós de árvore binária, respectivamente.

A seguir, está apresentado o Algoritmo 1 na próxima página proposto, em que λ representa o nó vazio. Com base nessas informações e supondo que a raiz de uma árvore binária com n nós seja passada ao procedimento `PRÉ-ORDEM()`, julgue os itens seguintes.

- I. O algoritmo visita cada nó da árvore binária exatamente uma vez ao longo do percurso.
- II. O algoritmo só funcionará corretamente se o procedimento `DESEMPILHA()` for projetado de forma a retornar λ caso a pilha esteja vazia.
- III. Empilhar e desempilhar referências para nós da árvore são operações que podem ser implementadas com custo constante.

Algoritmo 1:**Algoritmo:** PRÉ-ORDEM(*ptr*_{raiz})**Entrada:** nó de árvore binária *ptr*_{raiz}*ptr* ← *ptr*_{raiz}**enquanto** *ptr* ≠ λ **faça** PERCORRER(*ptr*) **se** *ptr.dir* ≠ λ **então** EMPILHAR(*ptr.dir*) **se** *ptr.esq* ≠ λ **então** EMPILHAR(*ptr.esq*) *ptr* ← DESEMPILHAR()IV. A complexidade do pior caso para o procedimento PRÉ-ORDEM() é $O(n)$.

Assinale a opção correta.

- A. Apenas um item está certo.
- B. Apenas os itens I e IV estão certos.
- C. Apenas os itens I, II e III estão certos.
- D. Apenas os itens II, III e IV estão certos.
- E. Todos os itens estão certos.

2 Exercícios de Aplicação

10. Escreva um algoritmo que constrói uma ABB de altura mínima a partir de um vetor ordenado V .
11. Escreva um algoritmo que, dada uma ABB T e um valor x , imprime todos os valores em T que sejam maiores ou iguais a x . Suponha que o algoritmo IMPRIME(v) seja responsável por imprimir elementos de T .

OBS: Perceba que não é necessário passar por todos os nós de uma ABB para realizar esse processo.

12. Considere uma *árvore binária de busca* com a seguinte modificação: em vez de cada nó armazenar diretamente o valor do elemento colecionado, armazenamos duas informações que chamaremos de *base* e *aditivo*. Para recuperarmos o valor do elemento armazenado em um nó, devemos partir do valor *base* desse nó e ir acumulando os *aditivos* de todos os seus ascendentes, além do seu próprio.

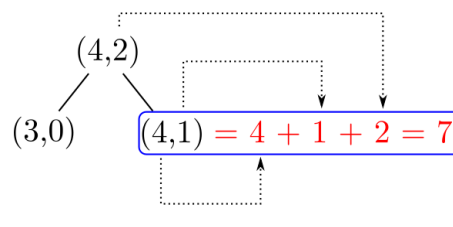
Por exemplo, a árvore na Figura 4 armazena os elementos 5, 6 e 7, supondo que os pares ordenados estão no formato (*base*, *aditivo*) para cada nó.

Figura 4:

- (a) Quais são todos os elementos colecionados na árvore da Figura 5 na próxima página?
 - (b) Escreva um algoritmo de busca para uma árvore com essa nova definição de nó.
13. Dada uma ABB T e um natural positivo k , escreva um algoritmo que retorna o nó onde encontramos o k -ésimo menor elemento de T .

OBS: A versão recursiva desse algoritmo é mais simples.

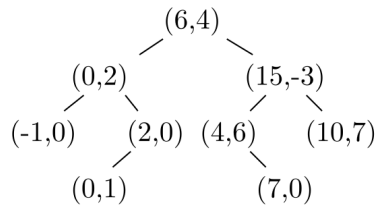


Figura 5:

14. Altere as informações que armazenamos nos nós de uma ABB de forma que seja possível encontrar seu k -ésimo menor elemento em $O(h)$. Explique como devemos alterar os algoritmos de ABB para manter essas informações consistentes.

3 Desafios

15. Dada uma ABB cheia T e um valor $x \in T$, imaginando que os nós foram numerados da esquerda para a direita, um nível por vez, determinar em $O(h)$ qual foi o número associado ao nó que contém x .
16. Escreva um algoritmo que, dada uma árvore binária de busca e um valor x pertencente à árvore, constrói uma segunda árvore binária de busca com todos os elementos menores que x .
17. Escreva um algoritmo que, dada uma ABB T e dois valores x e y , imprime todos os valores em T que estejam no intervalo $[x, y]$. Suponha que o algoritmo `IMPRIME(v)` seja responsável por imprimir elementos de T .
18. Dada uma ABB T e um valor x , alterar T e retornar outra ABB V de forma que em T permaneçam os valores menores que x e em V sejam postos os maiores ou iguais a x .
19. Projete uma estrutura de nó para ABBs de forma que a ordem de inserção seja preservada. (Ou seja, a qualquer momento podemos chamar `OBTERORDEMDEINSERÇÃO()` e obter uma lista com os nós na ordem em que foram inseridos. Imaginar que não permitimos remoções na árvore. Complexidade adicional $O(1)$). Não guarde os elementos duas vezes.
20. Escreva um algoritmo recursivo que recebe uma Árvore Binária de Busca e reorganiza as referências dos nós de forma a convertê-la em uma Lista Duplamente Encadeada Circular ordenada. A lista obtida deve ser formada dos mesmos nós da árvore passada, e não de cópias. A referência `esq` de cada nó deve apontar para o nó anterior na ordem e a referência `dir` deve apontar para o próximo, com exceção do primeiro e do último, que devem fechar o círculo. A ordem dos nós quando na lista deve ser não-decrescente quanto aos valores em chave. Retorne o primeiro nó da lista obtida.

Essa conversão pode ser realizada em $O(n)$, essencialmente operando uma vez sobre cada nó.