



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

LE5  
2025.1

Nome: \_\_\_\_\_ Matrícula: \_\_\_\_\_

## 1 Exercícios de Fixação

1. Explique a seguinte sentença: “Árvores  $B$  são construídas de baixo para cima, enquanto árvores binárias são construídas de cima para baixo”.
2. Considere a afirmação: “Uma árvore  $B$  não pode crescer em altura até que esteja 100% cheia”. Isso é verdade? Explique.
3. Mostre uma árvore  $B$  de ordem  $d = 2$  e altura 3 tal que a inserção de qualquer novo elemento implique no aumento da altura da árvore. Exiba a árvore após tal inserção.
4. A partir de uma Árvore  $B$  inicialmente vazia, com ordem  $d = 2$ , realize as inserções dos seguintes valores, na ordem em que aparecem:

$\langle 63, 16, 51, 77, 61, 43, 57, 12, 44, 72, 45, 34, 20, 7, 93, 29 \rangle$ .

**OBS:** Para facilitar e economizar tempo, redesenhe a árvore apenas após ser necessária uma cisão.

Agora, responda às seguintes questões:

- (a) Quantas inserções foram realizadas em nós que ainda tinham espaço disponível?
  - (b) Quantas cisões você precisou realizar?
  - (c) Qual é a altura da árvore obtida?
  - (d) Qual é a razão entre o número de espaços disponíveis e o número total de espaços da árvore obtida?
  - (e) Se inserirmos essas mesmas chaves em uma ordem diferente, é possível obter uma árvore que seja estruturalmente diferente? Se sim, indique em que ordem devemos inserir e desenhe a árvore final. Se não, explique por quê.
5. Suponha uma árvore  $B$ , inicialmente vazia, com ordem  $d = 2$ . Mostre como, estruturalmente, deverá ficar tal árvore após cada uma das operações na sequência a seguir, imaginando que  $I(x)$  representa uma operação de inserção do valor  $x$  na árvore e  $R(y)$  representa uma operação de remoção do valor  $y$  da árvore.  
 $\langle I(20), I(30), I(10), I(5), R(20), I(15), I(25), I(40), I(60), R(15), R(10) \rangle$
  6. Exiba uma sequência de 15 chaves que, se inseridas em uma Árvore  $B$  de ordem  $d = 2$ , inicialmente vazia, na ordem apresentada, gerarão o menor número possível de cisões dentre as outras ordenações para as mesmas chaves. Explique por que podemos nos convencer de que o número de cisões obtido é mínimo.
  7. Argumente para mostrar que se uma chave não está em um nó folha, seu sucessor, caso exista, deve estar.
  8. Que prejuízos e que vantagens obteríamos se substituíssemos a organização de cada nó de uma árvore  $B$  na forma de vetor pela organização em ABB? Em outras palavras, os elementos de cada nó sendo guardados em uma ABB em vez de em um vetor.

## 2 Exercícios de Aplicação

9. Escreva um algoritmo que, acessando cada nó de uma árvore B apenas uma vez, imprime *todos* os elementos da árvore em ordem *crescente*.
10. Escreva o algoritmo  $B\text{-}K\text{MAIORES}(T, k)$ , que dada uma Árvore B  $T$  e um natural  $k$  retorna as  $k$  maiores chaves presentes em  $T$ .
11. Escreva o algoritmo  $B\text{-}INTERVALO(T, x, y)$ , que dada uma Árvore B  $T$  e dois elementos  $x$  e  $y$ , com  $x \leq y$ , imprime todas as chaves em  $T$  que estejam dentro do intervalo  $[x, y]$  sem obrigatoriamente passar por todas as chaves de  $T$ .
12. Escreva um algoritmo  $BUSCAORDEM(T, k)$  tal que, dada uma Árvore B  $T$  e um natural não-nulo  $k$ , retorne o  $k$ -ésimo menor elemento naquela árvore. Considere que armazenamos em cada nó um terceiro vetor,  $T$ , onde  $T[i]$  representa quantos elementos residem na sub-árvore encontrada em  $P[i]$ .

## 3 Desafios

13. Escreva um algoritmo para, dada uma Árvore B  $T$  e uma chave  $x$ , sabendo que  $x \in T$ , retornar o antecessor de  $x$  em  $T$ . Observe que nem sempre o que o algoritmo  $BUSCAB$  retorna é o antecessor de  $x$  com relação à árvore toda.