

Estruturas de Dados
Lista de exercícios 2

1. Faça um algoritmo que receba uma string e monte uma árvore geral onde cada nó tem um valor (dado por uma letra), um ponteiro para o pai, um ponteiro para o filho esquerdo e um ponteiro para o irmão direito. Na string, toda subárvore inicia com abre-parêntesis, seguido do valor do nó, seguido das subárvores filhas e terminando com fecha-parêntesis. Por exemplo, “(a (b(d)(e)) (c(f)(g)))”. Espaços na string podem ser ignorados e você pode assumir que a string de entrada virá sem erros de parêntesis.
2. Faça o mesmo algoritmo da questão anterior para uma árvore geral em que cada nó possui um valor (dado por uma letra), um ponteiro para o pai e um vetor de ponteiros para os filhos.
3. Seja T uma Árvore Binária de Busca inicialmente vazia. Desenhe a árvore T após cada uma das seguintes operações: Inserção de 10, 20, 30, 40, 50, 5, 15, 25, 35, 45, 1, 11, 21, 31, 41 e Remoção de 30, 10.
4. Inclua um campo “altura” em cada nó e faça alterações nas operações Insert e Delete das Árvores Binárias de Busca de modo a sempre ser possível saber a altura de qualquer nó da árvore. As operações devem continuar sendo em tempo $O(h)$, onde h é a altura da raiz da árvore.
5. Uma Árvore de Estatística de Ordem é uma variante das Árvores Binárias de Busca que contém as seguintes operações adicionais: Tree-Select(i) (encontra o i -ésimo menor elemento da árvore) e Tree-Order(x) (encontra a ordem em que se encontra um elemento com valor x). Para implementá-la, inclua um campo “tamanho” em cada nó para determinar o número de elementos da subárvore enraizada no nó (inclusive).
 - (a) Altere as operações Insert e Delete das Árvores Binárias de Busca para comportar esse novo campo.
 - (b) Escreva o algoritmo para a operação Tree-Select.
 - (c) Escreva o algoritmo para a operação Tree-Order.
6. Escreva a operação Right-Rotate em pseudo-código ou em C++: dado um nó u que tem filho esquerdo v em uma árvore binária de busca, fazer u ser filho direito de v , ajustando os possíveis filhos de v de modo a manter a propriedade de árvore de busca binária.
7. Considere a inserção da seguinte sequência de números em uma certa estrutura de dados inicialmente vazia: 6, 1, 2, 8, 5, 4, 4, 7, 9, 3. Desenhe a estrutura de dados resultante em cada um dos seguintes casos: (a) Árvore Binária de Busca e (b) Árvore AVL
8. Considere a seguinte sequência de operações em uma Árvore AVL inicialmente vazia: Inserção de 10, 20, 30, 40, 50, 5, 15, 25, 35, 45, 1, 11, 21, 31, 41 e Remoção de 30, 10. Desenhe **passo a passo** a Árvore AVL após cada operação.