

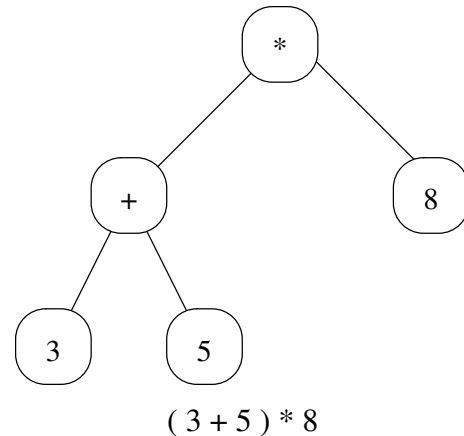
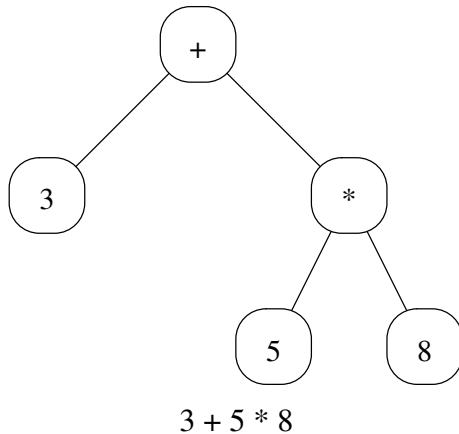


Lista de Exercícios sobre Árvore

- 1) Quantos antecedentes tem um nó no nível n em uma árvore binária? Prove sua resposta.
- 2) Uma árvore estritamente binária com n nós folhas contém quantos nós?
- 3) Escreva algoritmos recursivos e não-recursivos para determinar:
 - a. O número de nós em uma árvore binária
 - b. A soma dos conteúdos de todos os nós em uma árvore binária, considerando que cada nó contém um inteiro
 - c. A profundidade de uma árvore binária
- 4) Escreva um algoritmo para determinar se uma árvore binária é:
 - a. estritamente binária
 - b. completa
 - c. quase completa
- 5) Duas árvores binárias são **similares** se elas são vazias ou se elas não são vazias e suas subárvores da esquerda são similares e suas subárvores da direita são também similares. Escreva um algoritmo para determinar se duas árvores binárias são similares.
- 6) Duas árvores binárias são **espelho-similares** (*mirror similar*) se elas são vazias ou se elas não são vazias e as subárvores esquerdas de cada uma são espelho-similares as subárvores direitas da outra. Escreva um algoritmo para determinar se duas árvores binárias são espelho similares.
- 7) Escreva um algoritmo para determinar se uma árvore binária é ou não **similar** (veja o exercício 5) à alguma subárvore de outra árvore.
- 8) Escreva um algoritmo para determinar se uma árvore binária é ou não **espelho-similar** (veja o exercício 6) à alguma subárvore de outra árvore.
- 9) Considere uma árvore binária quase completa, onde cada nó é composto por um inteiro e ponteiros para as subárvores direita e esquerda. Construa uma função que receba um valor inteiro e o nó raiz da árvore binária quase completa e insira um novo nó (representado pelo valor inteiro recebido) na árvore de forma que ela continue sendo uma árvore binária quase completa.
- 10) Considere uma árvore binária qualquer, onde cada nó é composto por um inteiro e ponteiros para as subárvores direita e esquerda. Construa uma função que receba um valor inteiro e o nó raiz da árvore binária e, então, remova um nó que contenha o valor inteiro passado como parâmetro. Após a remoção do nó, a árvore binária deve ser

reconstruída, ou seja, caso o nó removido tenha filhos, os nós filhos deverão ser conectados aos pais ou entre eles mesmos.

- 11) Considere árvores binárias que representam expressões aritméticas (composta por operandos compostos por um único algarismo, operações de +, -, * e / e parênteses) como as apresentadas abaixo.



Escreva um algoritmo que receba um TAD representando tais árvores e retorne um *string* corresponde a versão infixa da expressão que contém somente aqueles parênteses que são necessários.

- 12) Escreva um algoritmo que receba uma expressão matemática (composta por operandos compostos por um único algarismo, operações de +, -, * e / e parênteses) representada por um *string* e retorne uma árvore binária representando esta expressão.
- 13) Dada uma árvore binária que represente uma expressão matemática, construa um algoritmo que apresente (imprima) a versão infixa (ou central) da expressão
- 14) Dada uma árvore binária que represente uma expressão matemática, construa um algoritmo que apresente (imprima) a versão posfixa (ou pós-ordem) da expressão.

Exercícios extraídos de (Referências)

[1] Aaron M. Tenenbaum, Yedidyah Langsam, Moshe J. Augenstein, *Estruturas de Dados Usando C*, Makron Books/Pearson Education, 1995.