ESCOLA POLITÉCNICA DE PERNAMBUCO – POLI
ENGENHARIA DA COMPUTAÇÃO - ECOMP
ESTRUTURAS DE DADOS - 2019.2

Lista de exercícios 05 - Árvores
27 de Outubro de 2019

OBS: Para cada questão solucionada, envie em seu projeto os testes efetuados em um método main.

(RESOLVA TODOS OS EXERCÍCIOS PROPOSTOS ABAIXO)

1) Implemente uma estrutura de dados que modela uma árvore de expressões, cujo construtor da classe recebe uma string de uma equação e monta a árvore de expressões correspondente. Assuma que a equação é formada apenas por números inteiros positivos ou negativos e que cada termo da equação é separado por um espaço em branco. Ex.: ((3 * 2) - ((4 / 2) + (-10 / 5)))
Dica: utilizar StringTokenizer.

2) Implemente na classe do exercício anterior o método que calcula o valor de uma árvore de expressões.

3) Adapte a implementação das questões 1 e 2 para permitir a construção de uma equação com variáveis. Para tanto, considere que no momento em que for acionado o método para cálculo da árvore de expressões, o programa perguntará ao usuário os valores de cada variável, devendo usar os valores informados para cálculo da expressão, sem que se modifique a equação original. Ou seja, o usuário poderá chamar o método calcular, quantas vezes desejar e em cada iteração novos valores das variáveis poderão ser informados pelo usuário.

4) A respeito das árvores binárias, implemente:

a) Escreva um algoritmo que conte o número de nós de uma árvore binária.

b) Escreva um algoritmo que conte o número de folhas de uma árvore binária.

c) Calcular a profundidade de um nó (passado como argumento) de uma árvore.

5) A respeito das árvores binárias que armazena valores inteiros, desenvolva os seguintes métodos:

a) Retornar uma lista encadeada com os elementos da árvore com os seguintes percursos: em ordem, pré-ordem e pós ordem.

b) Retornar uma lista encadeada com os valores dos nós de uma árvore binária que possuem a mesma altura e profundidade, simultaneamente.

6) Considere um TAD árvore binária ordenada de inteiros com seus métodos usuais. Desenvolva um método que recebe uma árvore e verifica se ela é AVL ou não. Caso não seja, o método realiza a conversão para AVL. O TAD árvore **não deve ser AVL por si só** (o add(T e) da árvore não faz essa verificação).

7) Implemente uma árvore binária e dois métodos: um para achar a folha mais longe da raiz, e outro para achar a mais próxima.

8) Desenvolva um método que recebe uma árvore binária de inteiros e retorna uma lista simplesmente encadeada, onde cada valor da lista é a soma de todos os valores de um nível n da árvore. No final, a lista terá x elementos, onde x é a quantidade de níveis da árvore.

9) Implemente uma árvore binária com os seguintes métodos:

a) numNosImpares(), que retorna a quantidade de nós com profundidade ímpar na árvore.
b) arvoreDif(Arvore arv1, Arvore arv2), que retorna uma árvore binária com os valores de “arv1” que não pertencem a “arv2”.

10) Implemente uma árvore binária com todos seus métodos usuais e desenvolva um método que recebe uma árvore e, caso ela esteja contida na árvore da classe, remova a árvore contida e os ramos que partem dela.