



Instituto Federal de Educação, Ciência e Tecnologia de Brasília – Câmpus Taguatinga  
Ciência da Computação – Estruturas de Dados e Algoritmos  
Lista de Exercícios – Árvores  
Prof. Daniel Saad Nogueira Nunes

Aluno: \_\_\_\_\_

Matrícula: \_\_\_\_\_

### Exercício 1

Em uma árvore elabore um algoritmo que efetue as seguintes buscas imprimindo os seus valores:

- Busca em Largura.
- Busca em profundidade em pré-ordem.
- Busca em profundidade em ordem.
- Busca em profundidade em pós-ordem.

### Exercício 2

Qualquer árvore pode ser representada via uma sequências de parênteses balanceados. Esta sequência é obtida utilizando o seguinte procedimento:

- Se o nó é **NULL**, imprima “()”.
- Ao chegar em um nó, imprima ‘(’.
- Chame o procedimento recursivamente para o filho da esquerda.
- Chame o procedimento recursivamente para o filho da direita.
- Imprima ‘)’.

A Figura 1 ilustra este processo.

Elabore um algoritmo que receba uma árvore e imprima a sequência de parênteses balanceados.

### Exercício 3

Faça um algoritmo que leia de um arquivo uma linha contendo a sequência de parênteses balanceados e na outra o valor de cada nó de acordo com esta mesma sequência e construa uma árvore.

A Figura 2 fornece um exemplo de entrada e saída.

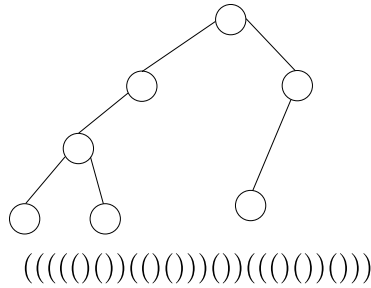


Figura 1: Conversão topologia  $\Rightarrow$  parênteses balanceados.

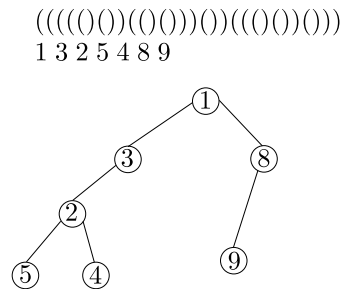


Figura 2: Conversão parênteses balanceados  $\Rightarrow$  árvore.

#### Exercício 4

Dada uma árvore binária  $T$ , desenvolva uma função que retorne a quantidade de folhas de  $T$ .

#### Exercício 5

Dadas árvores binárias  $T_1$  e  $T_2$ , escreva uma função que retorne verdadeiro se  $T_1$  é igual a  $T_2$  e falso caso contrário. Assuma que as árvores tem o mesmo tipo.

#### Exercício 6

Qual a maior e menor quantidade de nós que podemos ter em uma árvore binária completa de altura  $h$ ?

#### Exercício 7

Uma árvore binária tem a propriedade “zig-zag” quando ela é vazia ou não possui nós com dois filhos. Escreva um algoritmo que recebe uma árvore binária e determine se ela tem a propriedade “zig-zag”.

#### Exercício 8

Responda com verdadeiro ou falso e justifique a sua resposta:

- Qualquer que seja o número de chaves, é sempre possível construir com elas uma árvore binária completa.
- Qualquer que seja o número de chaves, é sempre possível construir com elas uma árvore binária cheia.
- Uma árvore binária que possui as folhas no último ou penúltimo níveis é completa.

---

## Exercício 9

Duas árvores  $T_1$  e  $T_2$  são ditas espelhadas, se elas são vazias ou se a subárvore da esquerda de  $T_1$  é espelhada em relação à subárvore da direita de  $T_2$  e se a subárvore da direita de  $T_1$  é espelhada em relação à subárvore da esquerda de  $T_2$ . Projete um algoritmo que receba duas árvores  $T_1$  e  $T_2$  e verifique se uma é espelhada em relação à outra.

## Exercício 10

Escreva um algoritmo para checar se uma determinada árvore binária é uma árvore binária de pesquisa (BST).

## Exercício 11

Rotações são operações essenciais no balanceamento de árvores binárias. Escreva os algoritmos de:

- $\text{LEFT-ROTATE}(x)$ : performa a rotação para esquerda no nó  $x$ .
- $\text{RIGHT-ROTATE}(x)$ : performa a rotação para direita no nó  $x$ .

## Exercício 12

Para árvores binárias de pesquisa (BSTs), escreva as funções de:

- Inserção.
- Remoção.
- Busca.

## Exercício 13

Para árvores AVL, escreva as funções de:

- Inserção.
- Remoção.
- Busca.

## Exercício 14

Para Treaps, escreva as funções de:

- Inserção.
- Remoção.
- Busca.

## Exercício 15

Um conjunto é um objeto matemático que corresponde a uma coleção de elementos distintos do mesmo tipo. Muitas linguagens de programação de alto nível possuem o objeto **conjunto**, o qual tem a capacidade inserir, remover e buscar elementos em tempo eficiente. Escolha a estrutura de dados apropriada para implementar o objeto **conjunto** e discorra o porquê desta estrutura é apropriada em relação às outras vistas no curso.

---

## Exercício 16

Implemente a estrutura de dados **conjunto** em C utilizando como base a estrutura escolhida no exercício anterior.

## Exercício 17

Um mapeamento consiste em levar objetos de um conjunto  $A$  de tipo  $x$  para objetos de outro conjunto  $B$  de tipo  $y$ . A única restrição é que um objeto do conjunto  $A$  pode ter apenas um único mapeamento para outro objeto do conjunto  $B$ . Muitas linguagens de programação de alto nível possuem o objeto **mapeamento** que tem a capacidade de criar um novo mapeamento, remover um mapeamento e buscar um mapeamento. Escolha a estrutura de dados apropriada para implementar o objeto **mapeamento** e discorra porque esta estrutura é apropriada em relação às outras vistas no curso.

## Exercício 18

Implemente a estrutura de dados **mapeamento** em C utilizando como base a estrutura escolhida no exercício anterior.