



## LISTA DE EXERCÍCIOS – ALGORITMOS E ESTRUTURA DE DADOS III

### Lista de Exercícios 9

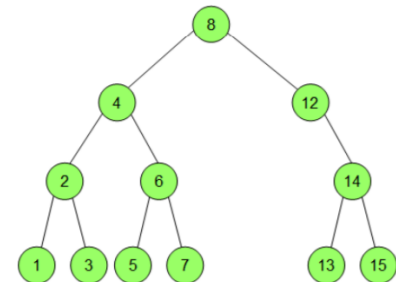
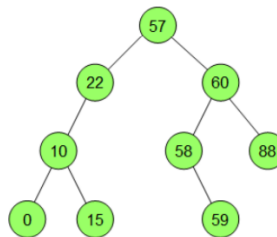
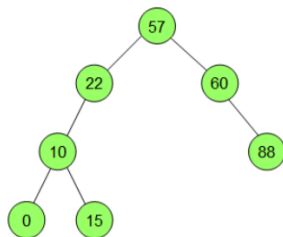
**DOCENTE:** Thiago França Naves

**DATA:** \_\_/\_\_/\_\_

**ALUNO:** \_\_\_\_\_

**OBS:** Todos os exercicios devem ser feitos utilizando o conceito de Árvores Binárias, com o projeto “ProjArvoreAVL” disponibilizado junto com o material da aula.

- 1) Examine cada árvore abaixo e responda se é ou não uma AVL e justifique a resposta.



- 2) Monte a árvore AVL (passo-a-passo) para as seguintes inserções de chaves, indicando a cada passo qual elemento foi inserido ou qual rotação foi realizada, ilustrar graficamente/textualmente como é feito o balanceamento:
- a) 50, 30, 20, 70, 40, 35, 37, 38, 10, 32, 45, 42, 25, 47, 36.
  - b) 100, 80, 60, 40, 20, 70, 30, 50, 35, 45, 55, 75, 65, 73, 77.
  - c) 41, 38, 31, 12, 19, 8, 27, 49.
  - d) 10, 21, 15, 17, 16, 19, 20
- 3) Um certo professor Amongus afirma que a ordem pela qual um conjunto fixo de elementos é inserido em uma árvore AVL não interessa – sempre resulta na mesma árvore. Apresente um pequeno exemplo que prove que ele está errado.
- 4) Construa a função de rotação simples a esquerda *void RotacaoRR(ArvAVL \*raiz)*.

- 5) Descreva passo a passo o funcionamento da função *insere\_ArvAVL(ArvAVL \*raiz, int valor)*, descreva também o uso de qualquer outras funções dentro dessa. Utilize trechos do código para exemplificar.
- 6) Descreva passo a passo o funcionamento da função *remove\_ArvAVL(ArvAVL \*raiz, int valor)*, descreva também o uso de qualquer outras funções dentro dessa. Utilize trechos do código para exemplificar.
- 7) Faça uma função que dada uma árvore verifica se a mesma é AVL.
- 8) Faça uma função *Arv\* transforma(Arv \*raiz)* que dada uma árvore binária de busca qualquer, retorna uma nova árvore AVL.