

# Algoritmos e Estruturas de Dados I

## Lista de Exercícios 5: Árvores AVL

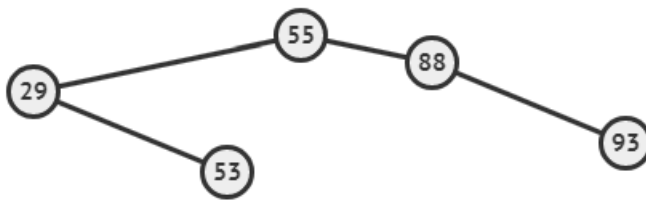
Profa. Mirtha Lina Fernández Venero

5 de abril de 2019

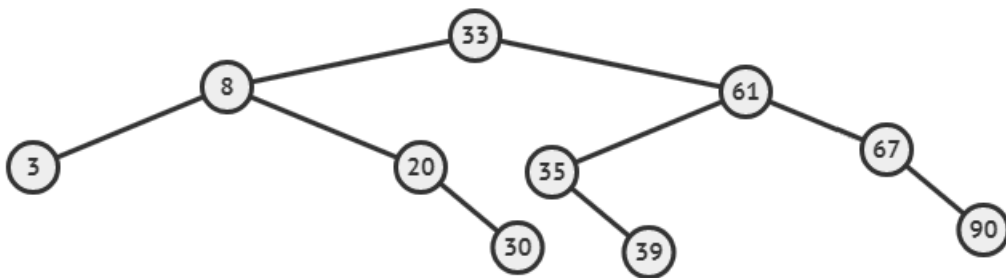
### 1 Árvores balanceadas

1. Quais das ABBs abaixo são AVL? Justifique de forma apropriada sua resposta

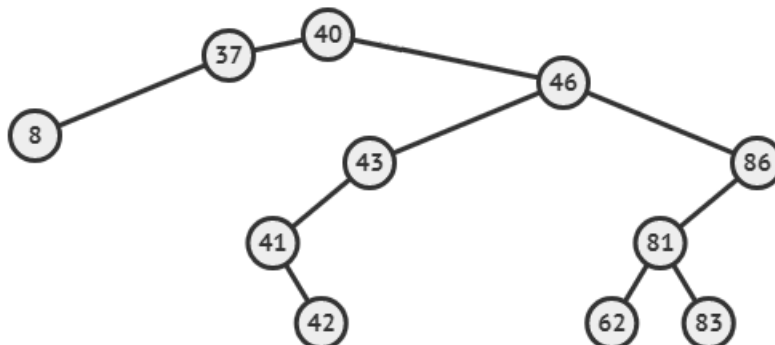
(a)



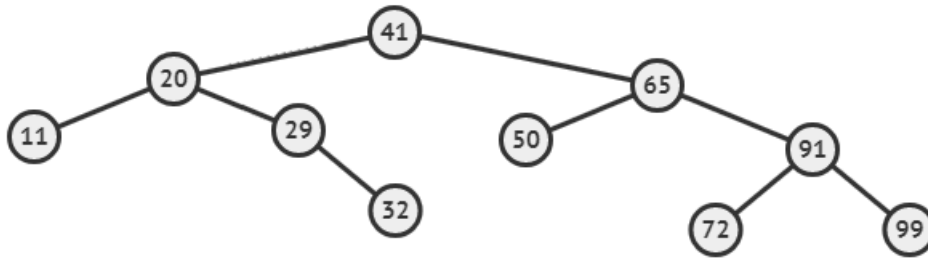
(b)



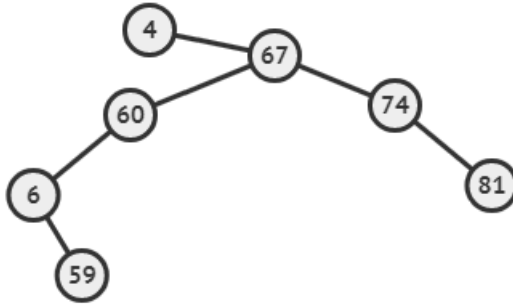
(c)



(d)



(e)



- Escreva um programa que leia  $n$  inteiros, construa uma ABB e determine se é ou não AVL. Caso não seja, o programa deve imprimir todas as chaves dos nós que não cumprem a propriedade
- Uma árvore binária se diz uniformemente podada se todas as folhas têm a mesma profundidade.
  - Quais das árvores acima são uniformemente podadas? Toda árvore uniformemente podada é AVL? Justifique suas respostas.
  - Escreva uma função que determine se uma árvore binária é uniformemente podada.

## 2 Rotações em árvores binárias

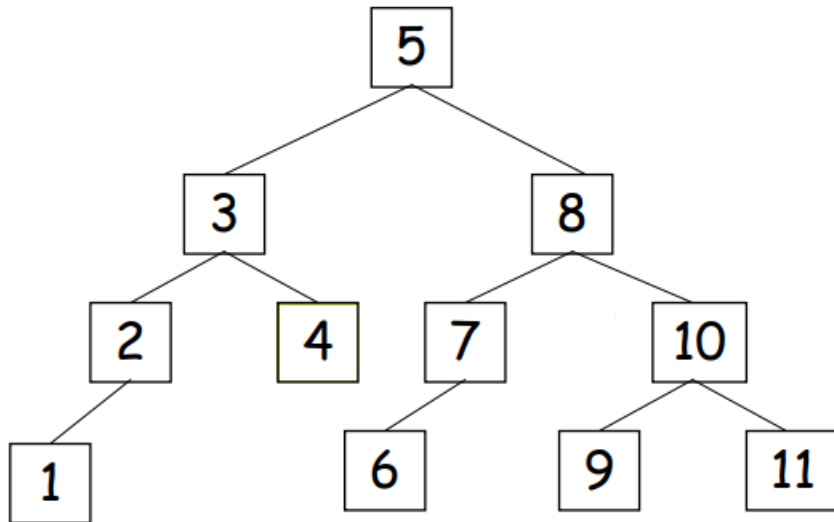
- Desenhe o resultado de realizar as seguintes rotações nas ABBs do exercício 1.
  - Rotação simples esquerda do nó 55 árvore 1a).
  - Rotação dupla direita do nó 55 da árvore 1a).
  - Rotação simples direita do nó 81 da árvore 1c).
  - Rotação dupla esquerda do nó 40 da árvore 1c).
  - Rotação dupla direita do nó 60 da árvore 1e) seguida de uma rotação dupla esquerda do nó 4.
- Escreva uma função que dada um nó de uma ABB e um inteiro de valor 1, -1, 2 ou -2 faça a rotação simples à esquerda (1), simples à direita (-1), dupla à esquerda (1) ou dupla à direita (-1) do nó. O nó não deve ser modificado caso a rotação requerida não for possível. Use o seguinte protótipo:

```
void rotate(TreeNode **node, int rotation);
```

- Escreva uma função que dada uma ABB  $A$  de inteiros e uma chave  $k$ , transforme  $A$  usando rotações do forma tal que  $k$  seja raiz. Note que se  $k$  não estiver em  $A$ , ela deve ser primeiro inserida e depois rotacionada uma ou várias vezes até ser raiz.

### 3 Inserção e remoção nas árvores AVL

- Desenhe passo a passo as árvores que resultam de inserir em uma árvore AVL vazia as seguintes chaves: 100, 80, 60, 40, 20, 70, 30, 50, 35, 45, 55, 75, 65, 73, 77. Indique em cada passo a rotação efetuada.
- Desenhe cada uma das árvores resultantes de inserir as seguintes chaves em uma árvore AVL (usando a ordem lexicográfica): **maio, março, novembro, agosto, abril, janeiro, dezembro, fevereiro, julho, junho, outubro, setembro**
- Desenhe cada uma das árvores resultantes de remover os seguintes nós (nessa ordem) da árvore AVL abaixo: 4, 6, 7, 3, 2, 1, 5. Em cada passo, mostre as rotações usadas.



- Escreva uma função para remover uma chave de uma árvore AVL.

### Contagem de Palavras

- Escreva um programa que dado um arquivo de entrada com um texto de tamanho arbitrário gere um outro arquivo de saída tal que cada linha do arquivo de saída tenha uma palavra do arquivo de entrada e o número de ocorrências dela no texto. O arquivo de saída deve estar ordenado por palavras. Assuma que o texto de entrada somente contem palavras em minúsculas separadas por espaço.

Por exemplo, se o arquivo de entrada contem o seguinte texto:

`the beatles were one of the most influential music groups of the rock era`

então o arquivo de saída deve conter o seguinte:

beatles 1  
era 1  
groups 1  
influential 1  
most 1  
music 1  
of 2  
one 1  
rock 1  
the 3  
were 1

7. Discorra sobre a complexidade no caso pior do seu programa.