

### 1) O que é árvore binária?

É um conjunto finito de elementos, nós, tal que ou é uma árvore vazia, ou há um nó raiz, especial, e os demais nós podem ser divididos em subconjuntos disjuntos, as subárvores à esquerda e à direita da raiz, sendo estas também árvores binárias.

### 2) O que é árvore binária de busca?

É uma árvore binária cujo valor do nó à esquerda da raiz é inferior ao valor desta, e o nó à direita da raiz é superior ao valor desta.

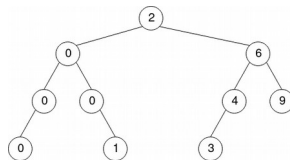
### 3) O que é árvore estritamente binária, completa e cheia?

Estritamente binária: nós possuem 0 ou 2 filhos.

Binária completa: todo nó que não possuir filhos deve estar ou no último ou no penúltimo nível da árvore.

Binária cheia: todo nó que não possuir filhos deve estar no último nível da árvore.

### 4) Faça uma árvore binária de busca com sua matrícula.



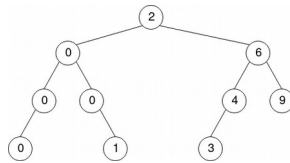
### 5) Quais os problemas de uma árvore binária de busca?

O problema é que, após muitas operações de inserção e remoção, estas podem se deformar, aumentando a complexidade da busca. Uma árvore binária de busca pode se deformar a tal ponto que se transforme em uma árvore zigue-zague, ou seja, praticamente uma lista encadeada.

### 6) O que é uma árvore balanceada?

É uma árvore binária de busca que, a despeito de inclusões ou remoções, mantém o custo de acesso em  $O(\log n)$ , ou seja, uma árvore de grandeza ótima.

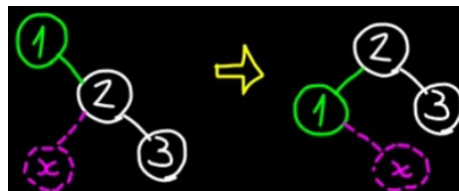
7) Faça uma árvore AVL com sua matrícula.



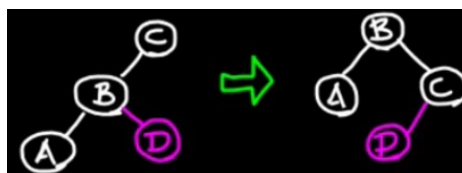
8) Explique o procedimento de rotação em uma árvore.

Há quatro tipos de rotação: à esquerda e à direita, e dupla à esquerda e dupla à direita.

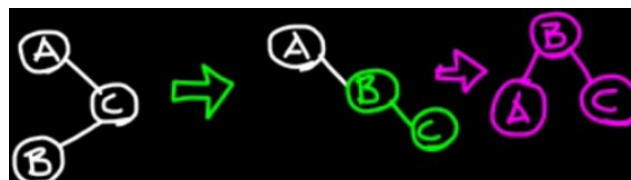
À esquerda:



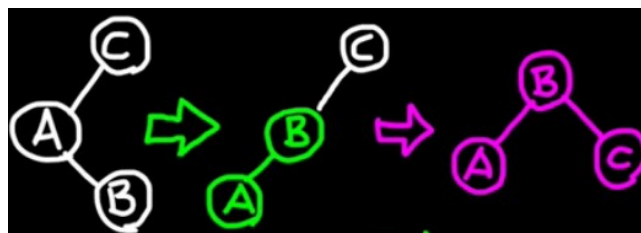
À direita:



Dupla à esquerda:



Dupla à direita:



### **9) Qual a justificativa para se usar árvores B?**

O fato de que, por manter mais de uma chave em cada nó da estrutura, a árvore B proporcionar uma organização de ponteiros de forma que operações de busca, inserção e remoção sejam executadas rapidamente. No mais, todas as folhas de uma árvore B encontram-se sempre no mesmo nível, não importando a ordem da entrada de dados.

### **10) O que a ordem significa numa árvore B?**

É por meio da ordem que se infere quantos filhos no máximo tem cada nó ( $2 * \text{ordem} + 1$ ), e também quantos filhos no mínimo estes nó, caso não sejam a raiz ou folhas, possuem ( $\text{ordem} + 1$ ).

---

## **NOTAS**

Para verificar se a árvore que você montou é uma binária de busca ou não, basta imprimi-la de forma simétrica, isto é, ir para o nó mais à esquerda, imprimir o nó, ir para o nó mais à direita, recursivamente.

Para verificar se uma árvore é AVL ou não, o índice de desequilíbrio de cada um de seus nós não pode ser menor que -1 ou maior que 1. O índice de desequilíbrio de um nó é calculado da seguinte maneira: para cada nó à esquerda deste nó raiz, subtraia 1; para cada nó à direita deste nó raiz, some 1. Como os nós-folhas não possuem nada à esquerda ou à direita, seu índice é sempre 0.

Para decidir qual rotação realizar em um nó AVL para manter a árvore organizada, deve-se seguir o algoritmo abaixo:

1. Calcular o índice de desequilíbrio do nó (índice chamado Q)
2. Se  $-1 \leq Q \leq 1$ , o nó está equilibrado
3. Se  $Q > 1$ :
  1. Se a subárvore da direita tem  $Q < 0$ :
    1. Rotação dupla à esquerda
  2. Se a subárvore da direita tem  $Q > 0$ :
    1. Rotação à esquerda
4. Se  $Q < -1$ :
  1. Se a subárvore da esquerda tem  $Q > 0$ :
    1. Rotação dupla à direita
  2. Se a subárvore da esquerda tem  $Q < 0$ :
    1. Rotação à direita