

Árvore AVL

A árvore AVL é bem parecida com a árvore binária de pesquisa porém ao inserir, também fazemos o balanceamento. Por isso precisamos da variável “nível” no seu construtor.

Como calculamos o nível? Precisamos mandar o nó em questão para o método, lá nós vamos usar verificar qual nível desse nó é maior, o direito ou o esquerda. Somamos 1 (do nível analisado) e obtemos o nível. É padrão ser direita - esquerda.

Como retornamos o nível? se for null recebe zero, caso contrário temos que calcular.

Vamos inserir normalmente e depois chamar o método balancear. Se o módulo diferença entre os níveis da esquerda e da direita (fator) for 1 ou 0, tá tudo ok.

Tipos de rotação: simples para esquerda ou para a direita, acontece quando o fator do neto do nó em questão é igual ao do pai.

Caso contrário os sinais sejam diferentes, a rotação será dupla, por que? Porque precisamos primeiro alinhar os 3 nós para depois rotacionar transformando o filho pai do avô e do neto.

Na rotação dupla: se quisermos rotacionar para a direita, primeiro vamos rotacionar para a esquerda e depois para a direita.

```
private No rotacionarDir(No no) {  
    System.out.println("Rotacionar DIR(" + no.elemento + ")");  
    No noEsq = no.esq;  
    No noEsqDir = noEsq.dir;  
  
    noEsq.dir = no;  
    no.esq = noEsqDir;  
  
    no.setNivel();  
    noEsq.setNivel();  
  
    return noEsq;  
}
```

```
private No rotacionarEsq(No no) {  
    System.out.println("Rotacionar ESQ(" + no.elemento + ")");  
    No noDir = no.dir;  
    No noDirEsq = noDir.esq;  
  
    noDir.esq = no;  
    no.dir = noDirEsq;  
  
    no.setNivel();  
    noDir.setNivel();  
    return noDir;  
}
```