



UNIVERSIDADE FEDERAL DO CEARÁ  
Campus de Quixadá  
Prof. Arthur Araruna  
QXD0115- Estrutura de Dados Avançada

# Árvores AVL são balanceadas

## 1 Conceitos

Vamos relembrar os nossos conceitos que serão necessários para o resultado a ser demonstrado:

**Definição 1.1.** Uma árvore binária  $T$ , com  $n$  nós, tal que  $h(T) = O(\lg n)$  é dita *balanceada*.

**Definição 1.2.** O *balanço* de um nó binário  $n$  é definido pela função  $bal(n) := h(n.dir) - h(n.esq)$ .

**Definição 1.3.** Uma árvore binária  $T$  é AVL quando, para todo nó  $n \in T$ ,  $|bal(n)| \leq 1$ .

## 2 Demonstração

### Observação:

A argumentação abaixo não é baseada na que consta em Szwarcfiter e Markenzon (2010) pois uma das afirmações dadas como válidas não é de fácil demonstração.

Note que, se mostrarmos que a altura de uma árvore AVL qualquer sempre é dada no máximo por uma função logarítmica do seu tamanho, poderemos concluir que a ordem de grandeza da altura não pode ser maior que logarítmica. Isso é exatamente o que significa, considerando  $h$  sendo a altura e  $n$  sendo o tamanho, que  $h = O(\lg n)$  e, com isso, que toda AVL seria balanceada.

Nesse sentido, uma das formas possíveis é se conseguirmos construir uma desigualdade da forma  $h \leq c \cdot k + \lg n$  para constantes  $c$  e  $k$ , com  $c > 0$ . A estratégia que seguiremos será um pouco mais restrita, porém permitirá concluirmos a mesma coisa.

Em vez de analisarmos uma árvore qualquer, vamos observar, para um dado  $h$  arbitrário, uma árvore AVL com altura  $h$  que contenha o mínimo possível de nós. Se essa árvore respeitar a desigualdade anterior, poderemos concluir que *todas* as AVL com essa mesma altura também a respeitarão (já que para essa tal árvore o lado direito da desigualdade é o menor possível, todas as outras apresentarão valores ainda maiores, o que mantém a desigualdade).

### 2.1 Estrutura da árvore

Para entendermos como construir uma árvore como desejamos, veja que sempre que, em uma árvore AVL  $T$  com altura  $h$ , encontrarmos um nó  $n$  com  $bal(n) = 0$ , poderíamos substituir uma de suas sub-árvores por outra AVL com uma altura 1 unidade menor que a que foi retirada sem invalidar a propriedade AVL de  $T$ . Isso constrói uma nova árvore  $T'$  com a mesma altura que a anterior  $T$  e certamente tamanho menor. Dessa forma, podemos concluir que a árvore desejada não pode ter nenhum nó com balanço nulo.

### 2.2 Contagem de nós da árvore

Vamos definir a função  $N(h)$  de forma a representar o menor número de nós possível para uma árvore AVL de altura  $h$ , que chamaremos de  $T$ .

Como vimos anteriormente, as sub-árvores de  $T$  devem possuir alturas  $h-1$  e  $h-2$ , não necessariamente nesta ordem. Além disso, podemos afirmar que ambas se tratam de árvores AVL com o mínimo de nós para a respectiva altura, caso contrário não faria sentido  $T$  possuir tamanho mínimo (já que elas poderiam ser trocadas por outras árvores de mesma altura e menos nós).

Dessa forma, podemos afirmar que o valor de  $N(h)^{(i)}$  pode ser calculado como

$$N(h) = \begin{cases} N(h-1) + N(h-2) + 1, & \text{se } h \geq 2; \\ 2, & \text{se } h = 2; \\ 1, & \text{se } h = 1. \end{cases}$$

Observe que a função  $N(h)$  é monotonicamente crescente<sup>(ii)</sup>. Com isso, e observando os casos-base, podemos afirmar que  $N(h) \geq 2N(h-2)^{(iii)}$  para todo  $h \geq 3$ .

### 3 Conclusão

Resolvendo<sup>(iv)</sup> a nova equação de recorrência obtida, obtemos que  $N(h) \geq 2^{\frac{h-1}{2}}$ . Lembrando que  $N(h)$  é o número de nós de  $T$ , que passaremos a denominar por  $n$ , concluímos que, para todo  $h \geq 3$ ,

$$\begin{aligned} n = N(h) &\geq 2^{\frac{h-1}{2}} && \implies \\ \lg n &\geq \frac{h-1}{2} && \implies \\ h &\leq 1 + 2 \lg n \end{aligned}$$

Dessa forma, temos todas as informações necessárias para concluirmos que  $h = O(\lg n)$ .

## Apêndice

### 4 Resolução da Recorrência

Na Seção 2.2 na página anterior determinamos que  $N(h) \geq 2N(h-2)$  para todo  $h \geq 3$ . Vamos agora mostrar que isso implica em  $N(h) \geq 2^{\frac{h-1}{2}}$  para todo  $h \geq 3$ .

Observe a seguinte sequência de desigualdades, obtida por sucessivas aplicações da desigualdade mencionada acima.

$$\begin{aligned} N(h) &\geq 2N(h-2) && = 2^1 N(h-2) \\ &\geq 2(2N(h-4)) && = 2^2 N(h-4) \\ &\geq 2^2(2N(h-6)) && = 2^3 N(h-6) \\ &\geq 2^3(2N(h-8)) && = 2^4 N(h-8) \\ &\dots && \dots \\ &\geq 2^{k-1}(2N(h-2k)) && = 2^k N(h-2k) \end{aligned}$$

Considerando que a última substituição nos tenha trazido a um caso base, temos as seguintes possibilidades:

<sup>(i)</sup>O caso  $h = 0$  não é considerado, mas é trivial, dado que a única árvore com altura 0 é a árvore vazia, com 0 nós.

<sup>(ii)</sup>Isso porque todos os valores envolvidos são não-negativos e as operações realizadas são apenas adições.

<sup>(iii)</sup>Considere que substituímos o termo  $N(h-1)$  por um termo  $N(h-2)$  e o termo 1 por um termo 0. Em ambas as substituições, os novos valores são menores.

<sup>(iv)</sup>Veja argumentação no Apêndice.

**CASO  $h - 2k = 2$ :** podemos concluir que  $k = \frac{h-2}{2}$  e que

$$N(h) \geq 2^{\frac{h-2}{2}} N(2) = 2^{\frac{h-2}{2}} \cdot 2 = 2^{\frac{h}{2}} \geq 2^{\frac{h-1}{2}}.$$

**CASO  $h - 2k = 1$ :** podemos concluir que  $k = \frac{h-1}{2}$  e que

$$N(h) \geq 2^{\frac{h-1}{2}} N(1) = 2^{\frac{h-1}{2}} \cdot 1 = 2^{\frac{h-1}{2}}.$$

## Referências

SZWARCFITER, J.L.; MARKENZON, L. **Estruturas de dados e seus algoritmos (3a. ed.)**. [S.l.]: Grupo Gen - LTC, 2010. ISBN 9788521629948.