AED2 - Aula 04
Árvores AVL: altura máxima e remoção

árvore binária

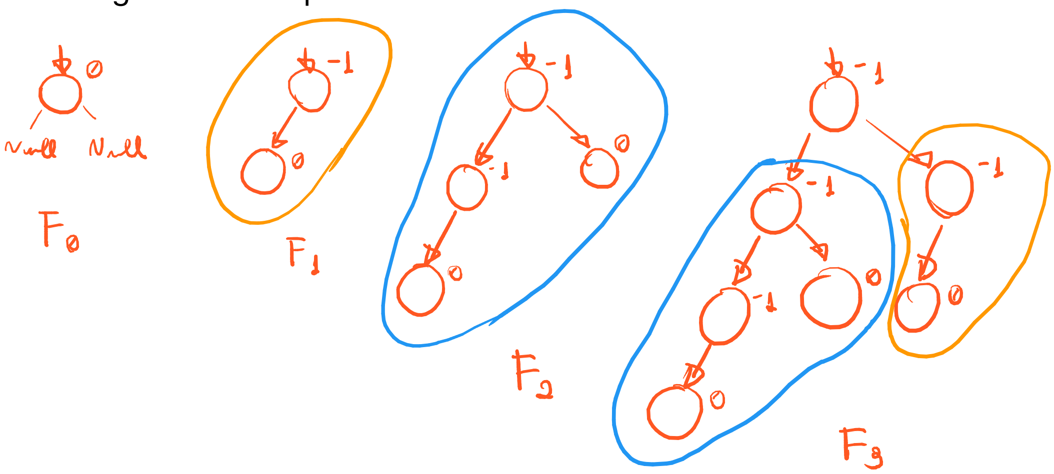
$\lfloor \lg n \rfloor \leq h \leq n-1$

Altura máxima de árvores AVL

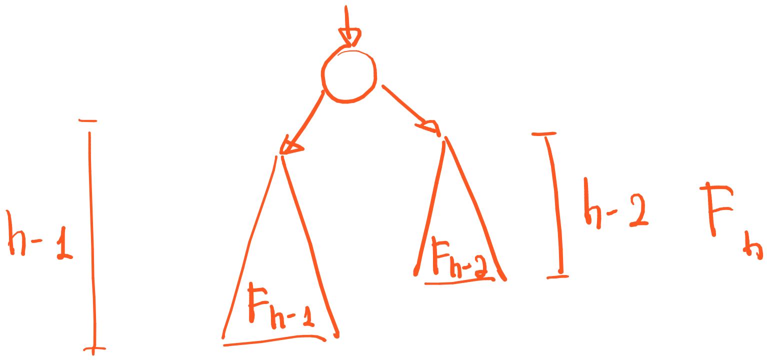
Quão esparsa pode ser uma árvore AVL?

- Isto é, dada uma árvore AVL de altura h,
    - qual o menor número de nós que ela pode ter?

Considere os seguintes exemplos:



Observe que, a regra recursiva de formação dessas árvores AVL esparsas/desbalanceadas é



- ou seja, F_h é composta por um nó raiz cujo
    - filho esquerdo é F_h-1
    - e filho direito é F_h-2.
- Ou seja, os filhos são árvores AVL com o menor número de nós possível.

Seja N(h) o número de nós da árvore F_h. Temos que

- N(0) = 1
- N(1) = 2
- Para h >= 2, N(h) = $1 + N(h-1) + N(h-2)$ (Recorrência)
    - Note que, N(h) é o menor número de nós que
        - uma árvore AVL de altura h pode ter.

Sendo N(h) = N(h - 1) + N(h - 2) + 1 para h >= 2, vamos expandir essa recorrência

- N(0), N(1), N(2), N(3), N(4), N(5), N(6), ...
- 1, 2, 4, 7, 12, 20, 33, ...
  - Enxergam um padrão?

Vamos comparar com a expansão da sequência de Fibonacci

- Fib(0), Fib(1), Fib(2), Fib(3), Fib(4), Fib(5), Fib(6), Fib(7), Fib(8), ...
- 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

Comparando as sequências podemos perceber que

- N(h) = Fib(h + 2) - 1, ☆
  - o que pode ser provado usando indução matemática.

Mas, Fib(h + 2) >= 2^(h / 2).

- Para verificar isso, perceba que
  - Fib(h + 2) = Fib(h + 1) + Fib(h) >= $Fib(h) + Fib(h) = 2\,Fib(h)$
- Ou seja, a cada dois incrementos no índice h
  - o valor na sequência de Fibonacci pelo menos dobra.

Assim, seja n o número de nós de uma árvore AVL de altura h. Temos que

$$n \geq N(h) = Fib(h+2) - 1 \geq 2^{(h/2)} - 1$$

$$2^{(h/2)} \leq n+1 \Rightarrow h/2 \leq \lg(n+1) \Rightarrow h \leq 2\lg(n+1)$$

- Portanto, a altura de uma árvore AVL é no máximo 2 lg (n+1),
  - i.e., O(log n).

$$Fib(h+2) \geq 2^{(h/2)} = \left(2^{1/2}\right)^h \cong 1{,}44^h$$

Bônus:

- É possível fazer uma análise mais precisa
  - em que mostramos que Fib(h) >= 1,618^h,
    - valor que deriva da razão aurea.
- Usando esse limitante inferior mais preciso para Fib(h) temos

$$n \geq N(h) = Fib(h+2) - 1 \geq 1{,}618^{(h+2)} - 1$$

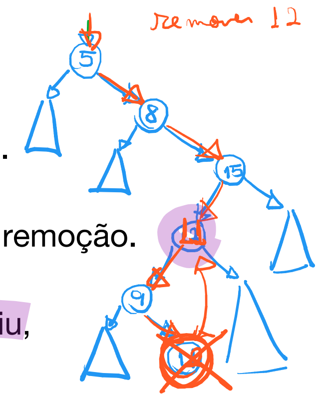
$$1{,}618^{(h+2)} \leq n+1 \Rightarrow h+2 \leq \log_{1{,}618}(n+1)$$

$$\Rightarrow h+2 \leq \lg(n+1)/\lg(1{,}618) \Rightarrow h \leq 1{,}44\lg(n+1) - 2$$

- Portanto,
  - $\boxed{h <= 1{,}44 \lg (n + 1) = O(\log n).}$

Remoção em árvores AVL

Similar aos casos da inserção, mas um tanto mais complexo.



Supomos que o algoritmo recursivo de remoção começa

- transformando, se necessário, a remoção do nó alvo
    - na remoção de um nó com no máximo um filho,
        - como ocorre nas árvores binárias de busca comuns.
- Então, analisamos o que precisa ser feito na volta da recursão,
    - quando a altura de uma das subárvores diminui após uma remoção.

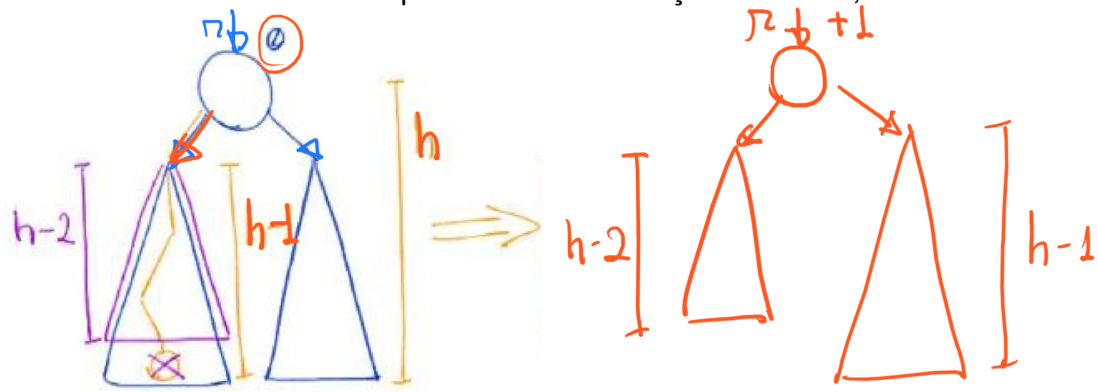
Caso 0: se a altura da subárvore em que ocorreu a remoção não diminuiu,

- o algoritmo não precisa realizar correções
- e devolve que a altura da sua árvore não diminuiu.

Caso 1: se o nó atual é o alvo, lembrando que ele tem no máximo um filho,

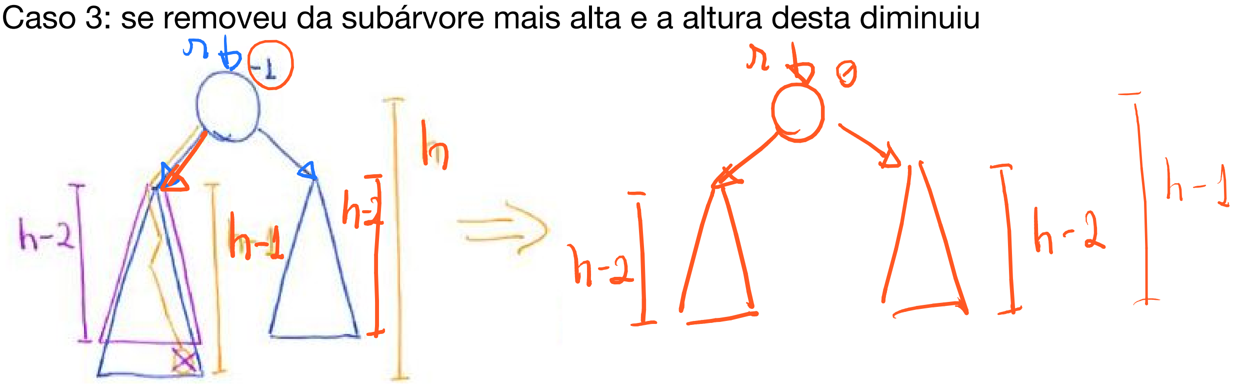
- remova o nó,
- corrija a subárvore resultante,
- e devolva que a altura diminuiu.

Caso 2: se a altura das duas subárvores era igual (i.e., balanceamento da raiz era 0) e a altura da subárvore em que ocorreu a remoção diminuiu,



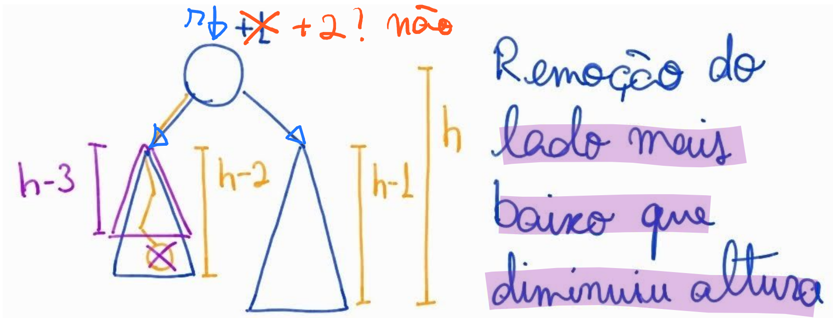
- corrige o fator de balanceamento
- e devolve que a altura da sua árvore não diminuiu.

Caso 3: se removeu da subárvore mais alta e a altura desta diminuiu



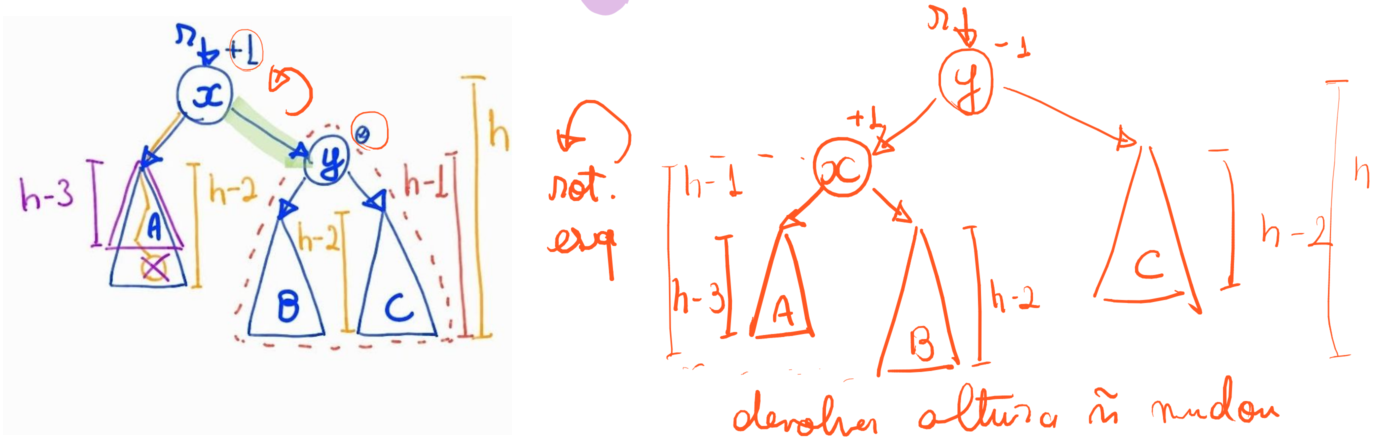
- corrige o fator de balanceamento para 0
- e devolve que a altura da sua árvore diminuiu.

Caso 4: se removeu da subárvore mais baixa e a altura diminuiu

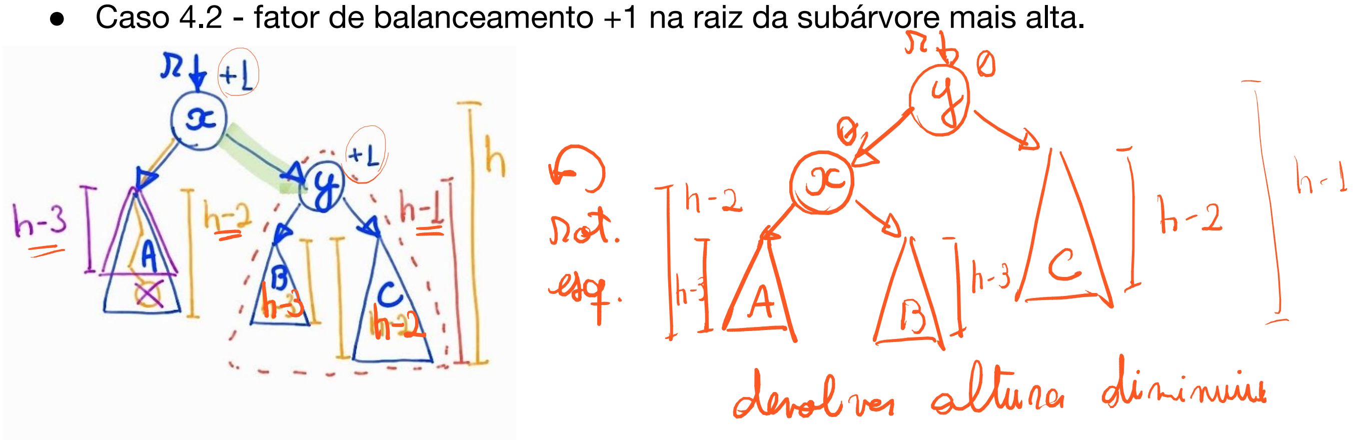


- é preciso realizar uma ou mais rotações para restaurar a propriedade AVL.

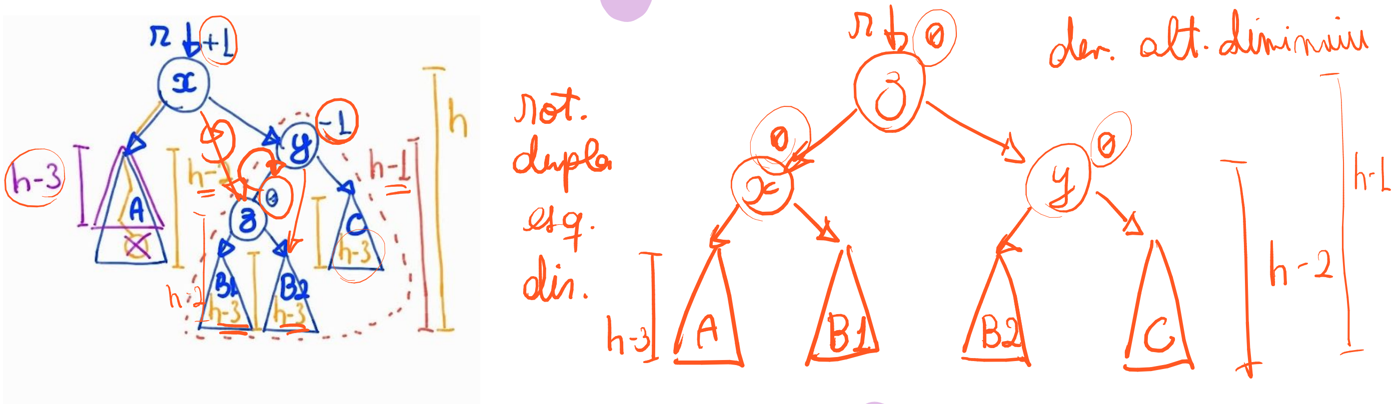
- Caso 4.1: fator de balanceamento 0 na raiz da subárvore mais alta.



- Caso 4.2 - fator de balanceamento +1 na raiz da subárvore mais alta.



- Caso 4.3 - fator de balanceamento -1 na raiz da subárvore mais alta.



  - Quiz: se o fator de balanceamento de z for -1/+1,
    - como ficam os balanceamentos de x e y após as rotações?

Note que, realizamos um número de operações constante por nível da árvore.

- Assim, a eficiência da remoção é proporcional à altura da mesma,
  - i.e., O(altura).