SIN5013 - Análise de Algoritmos e Estruturas de Dados

Árvores AVL

Prof. Flávio Luiz Coutinho
flcoutinho@usp.br

Árvores AVL: introdução

Trata-se de uma estrutura de dados inventada em 1962 por Adelson-Velsky e Landis. É uma especialização da árvore binária de busca, que possui como característica a garantia do seu balanceamento.

Árvores AVL: introdução

Sabemos que em uma árvore binária de busca, a complexidade computacional de da inserção, remoção ou busca é O(h), onde h é a altura da árvore.

Sabemos também que a altura de uma árvore que armazena n elementos pode variar em função da sequência de operações de inserção e remoção que foram realizadas previamente.

Árvores AVL: introdução

A cada inserção feita em uma árvore binária de busca, a altura h da árvore pode permanecer igual, ou aumentar em 1 unidade. No pior cenário possível que pode acontecer, todas n as inserções aumentam a altura h em 1 unidade. Neste cenário, teremos $h = \Theta(n)$.

Já no melhor cenário, os n elementos estarão distribuídos ao longo de k níveis completamente preenchidos da árvore. Neste caso teremos:

$$n = 2^0 + 2^1 + 2^2 + 2^3 + \ldots + 2^{k-1} = 2^k - 1 \longrightarrow n \sim 2^k$$

Se tomarmos a aproximação $n \sim 2^k$, então teremos que $k \sim \log_2 n$. Uma vez que a altura da árvore é dada por $h = k - 1$, então $h = \Theta(\log_2 n)$.

Árvores AVL: introdução

Embora se saiba que, na média, a altura de uma árvore binária de busca gerada aleatoriamente seja $O(\log_2 n)$, o que faz com que na média as operações de inserção, remoção e busca tenham complexidade $O(\log_2 n)$, é impossível garantir que qualquer árvore binária de busca tenha altura logarítmica.

Além disso, é difícil prever a altura de uma árvore após uma sequência de inserções e remoções.

Árvores AVL: introdução

Uma árvore AVL tem como objetivo, além de funcionar como uma árvore binária de busca, garantir o balanceamento da árvore.

Em uma árvore AVL, a diferença máxima de altura entre a subárvore à esquerda e a subárvore à direita será no máximo 1, para qualquer nó. Isso irá assegurar que a altura da árvore sempre seja $O(\log_2 n)$.

As operações de inserção e remoção de elementos são implementadas de modo a garantir esta propriedade.

Árvores AVL: balanceamento de um nó

Definiremos o balanceamento de um nó como sendo a diferença das alturas das subárvores direita e esquerda de um nó. Em uma árvore AVL, o balanceamento de um nó sempre será -1, 0 ou 1.

Árvores AVL: balanceamento de um nó

Definiremos o **balanceamento de um nó** como sendo a diferença das alturas das subárvores direita e esquerda de um nó. Em uma árvore AVL, o balanceamento de um nó sempre será -1, 0 ou 1.

Árvores AVL: inserção

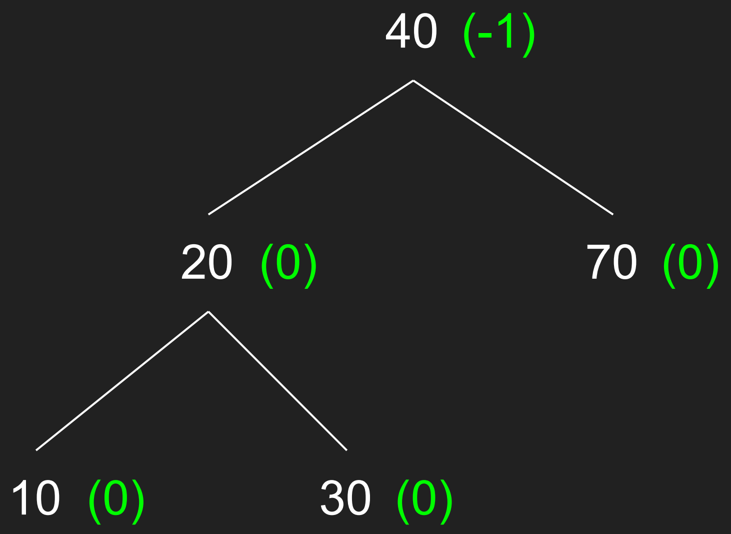
A inserção de um elemento em uma AVL é implementada de modo similar à inserção em uma árvore binária de busca, mas com alguns detalhes extras:

- atualiza-se o balanceamento (ou a altura) dos nós que fazem parte do “caminho” percorrido desde a raiz até o novo elemento inserido.
- se a propriedade do balanceamento de algum nó for violada em função da inserção do elemento novo, então é feita uma rotação para corrigir a árvore.

Observem que o nó recém inserido, assim como seu pai, nunca irão violar a regra do balanceamento.

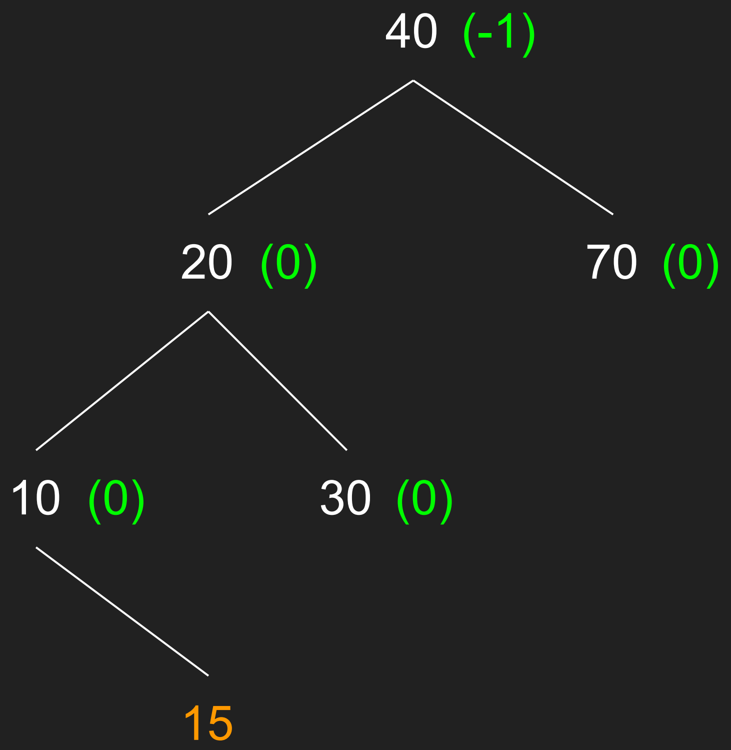
Árvores AVL: inserção

exemplo…



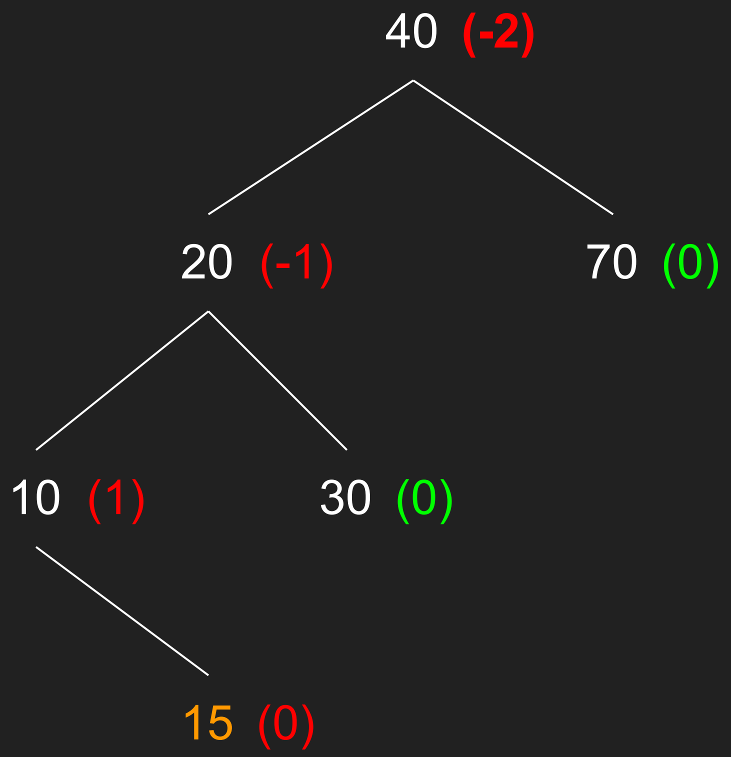
Árvores AVL: inserção

exemplo…



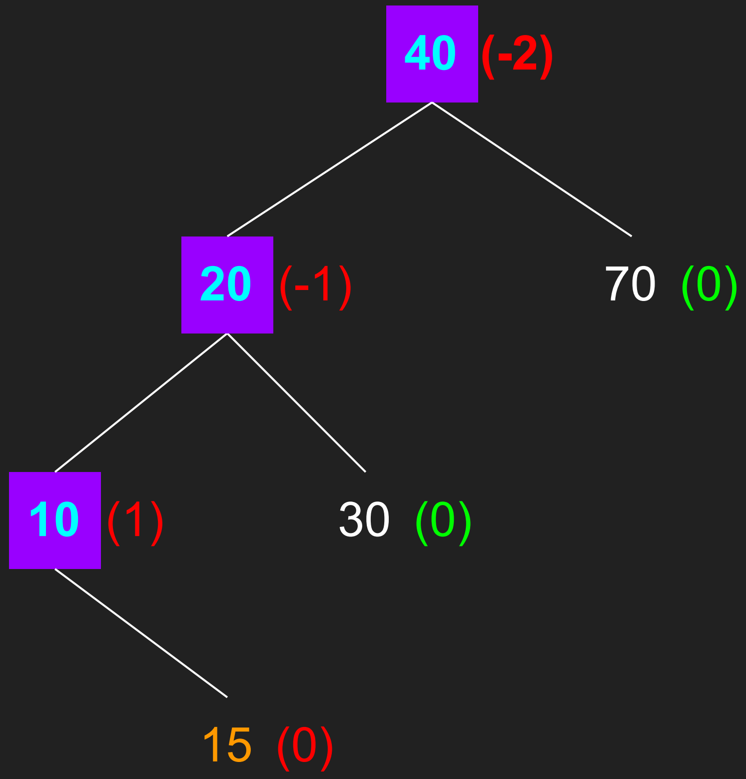
Árvores AVL: inserção

exemplo…



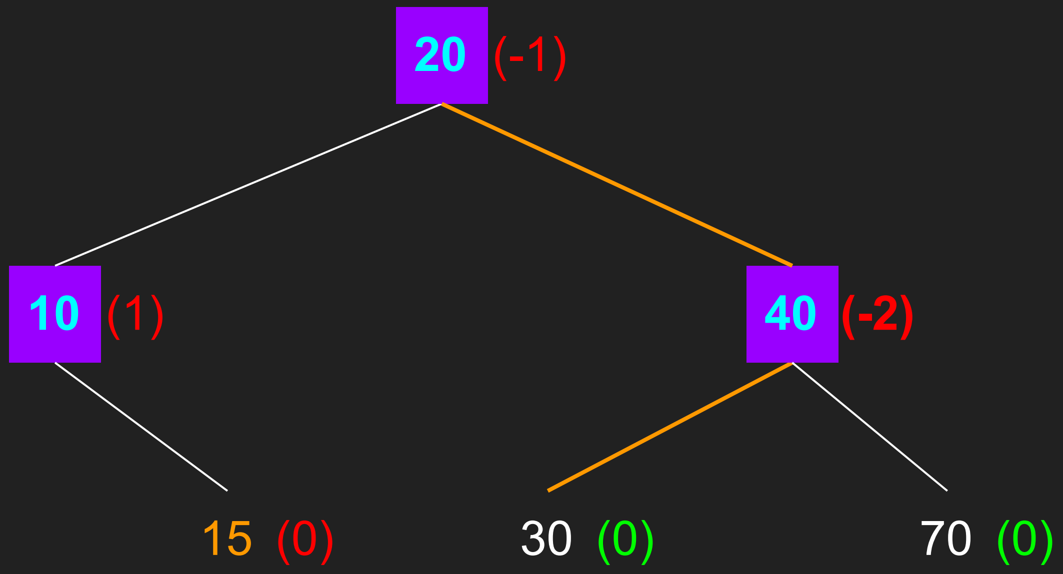
Árvores AVL: inserção

exemplo…



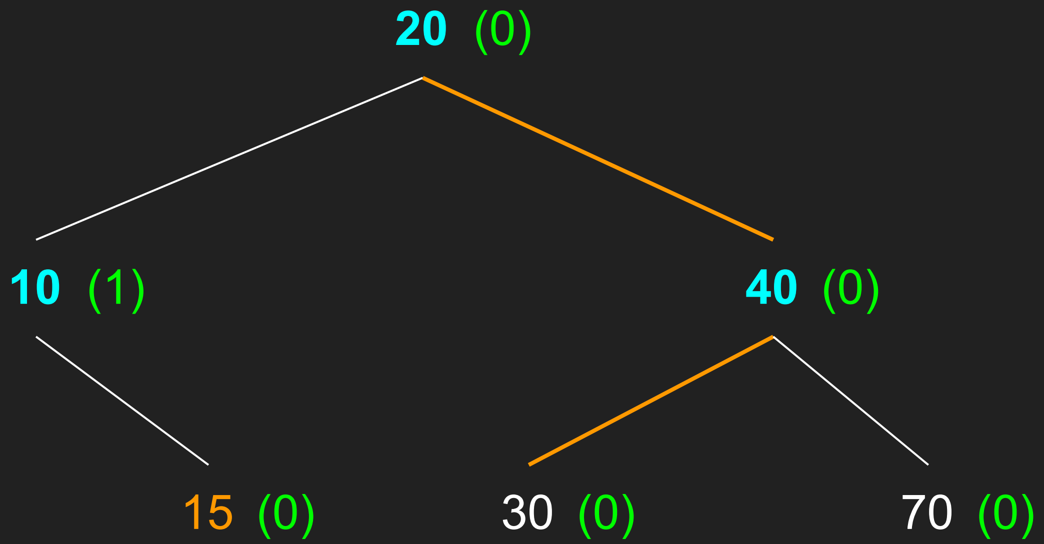
Árvores AVL: inserção

exemplo…



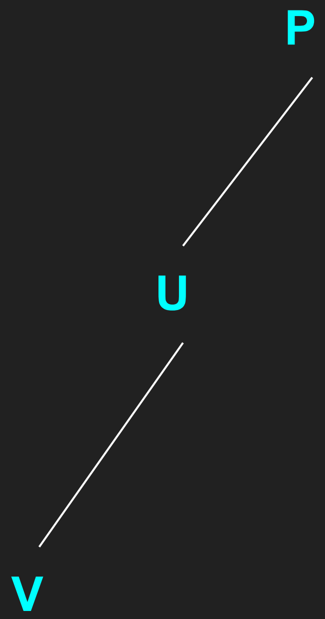
Árvores AVL: inserção

exemplo…



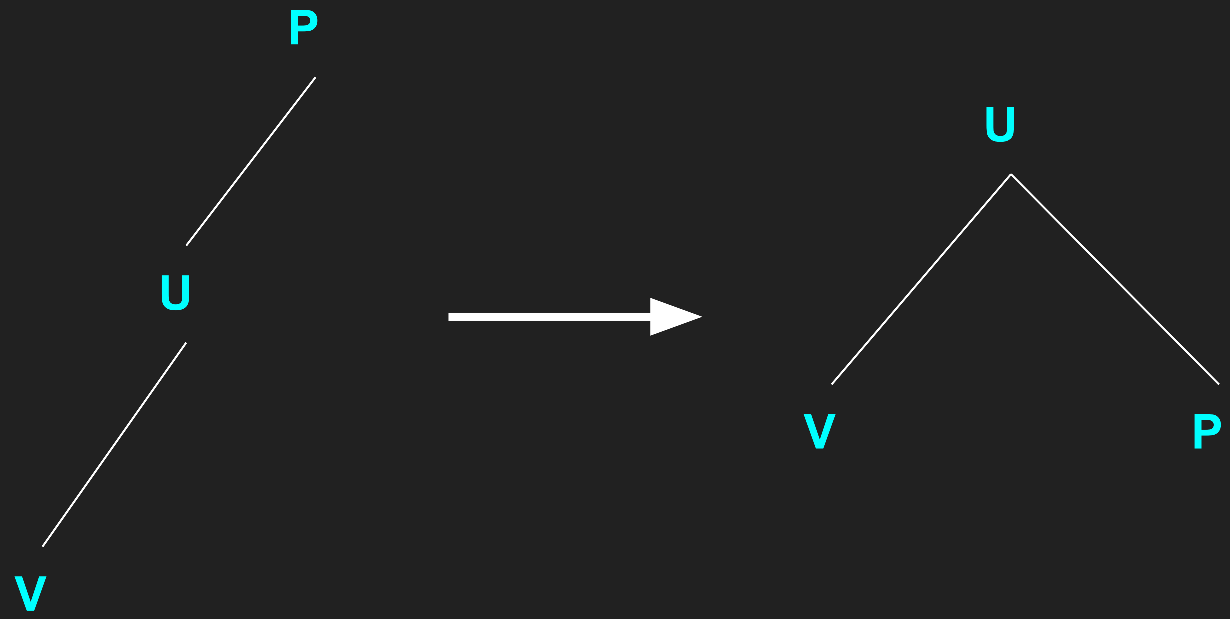
Árvores AVL: inserção

generalizando (LL)



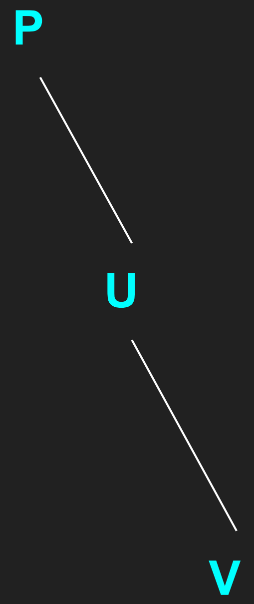
Árvores AVL: inserção

generalizando (LL)



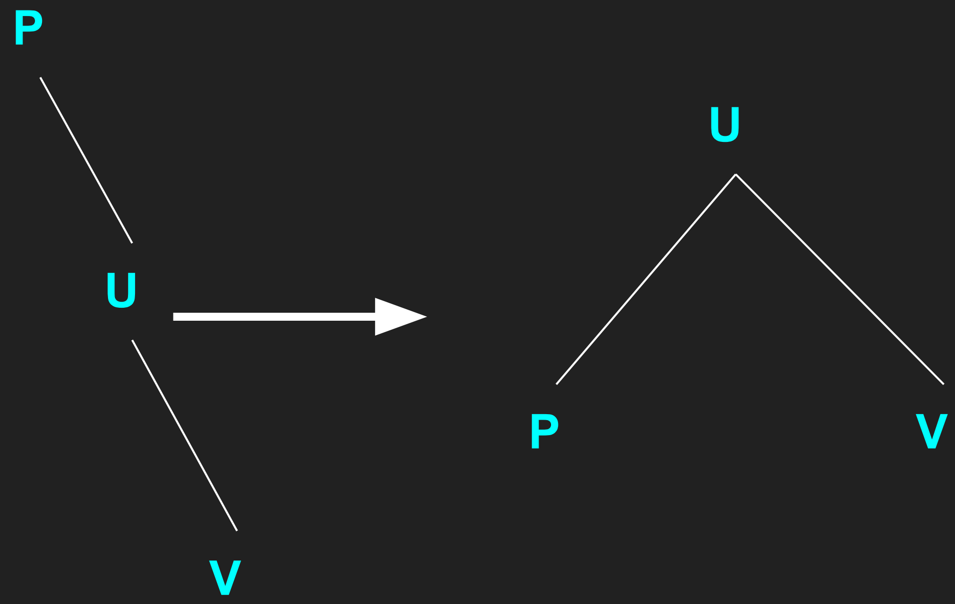
Árvores AVL: inserção

generalizando (RR)



Árvores AVL: inserção

generalizando (RR)

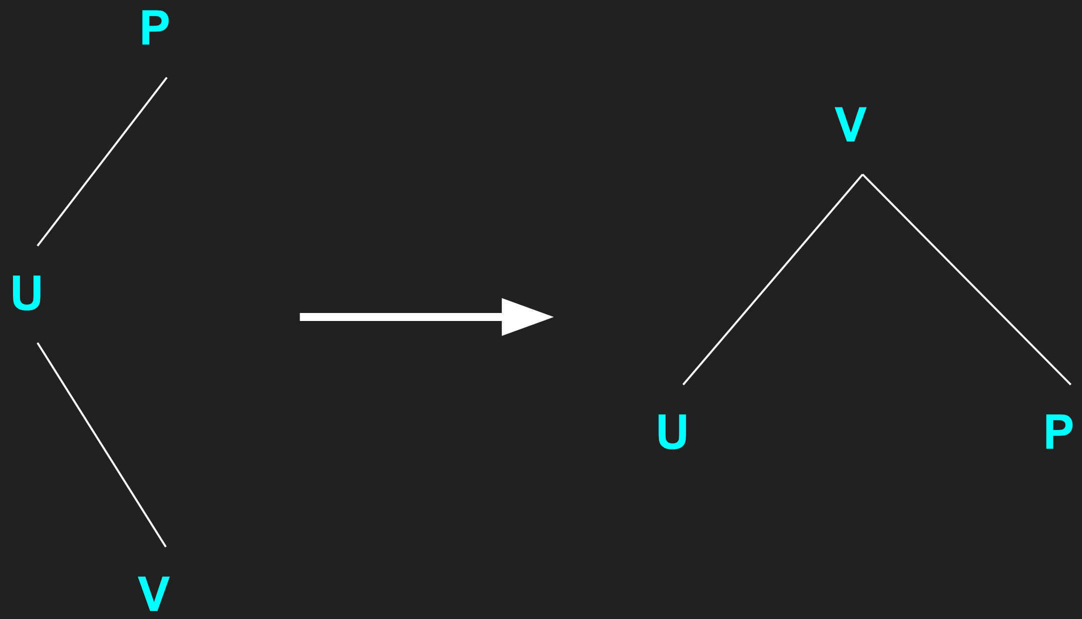


Árvores AVL: inserção

generalizando (LR)

Árvores AVL: inserção

generalizando (LR)

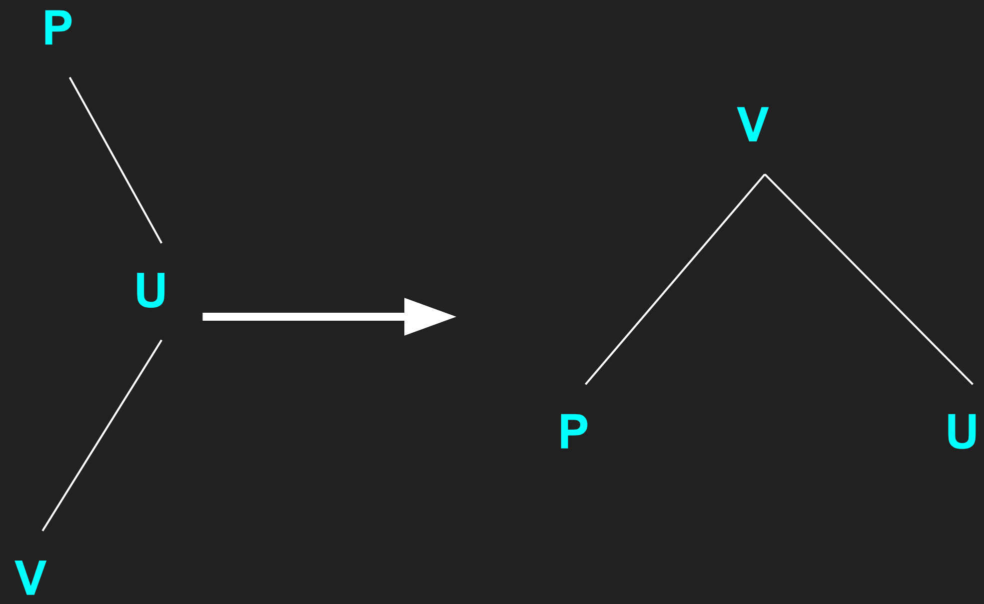


Árvores AVL: inserção

generalizando (RL)

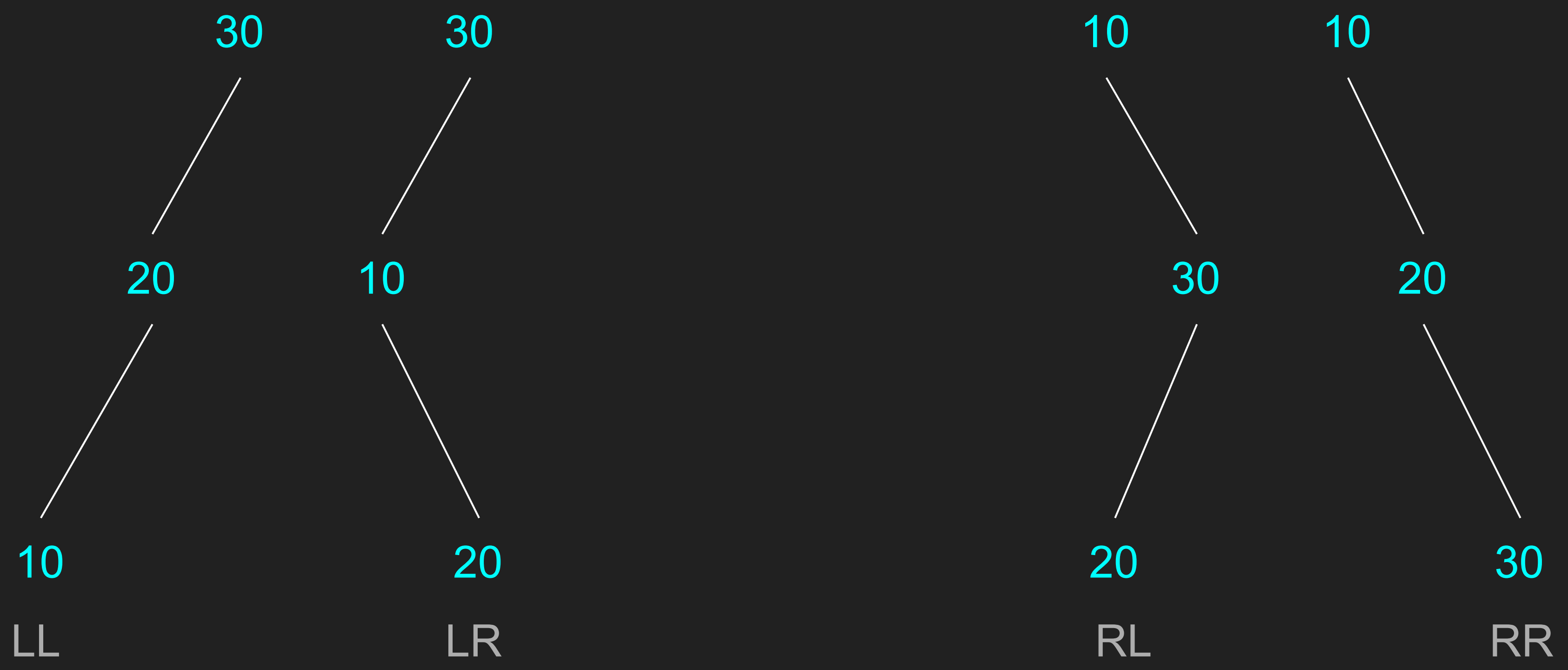
Árvores AVL: inserção

generalizando (RL)



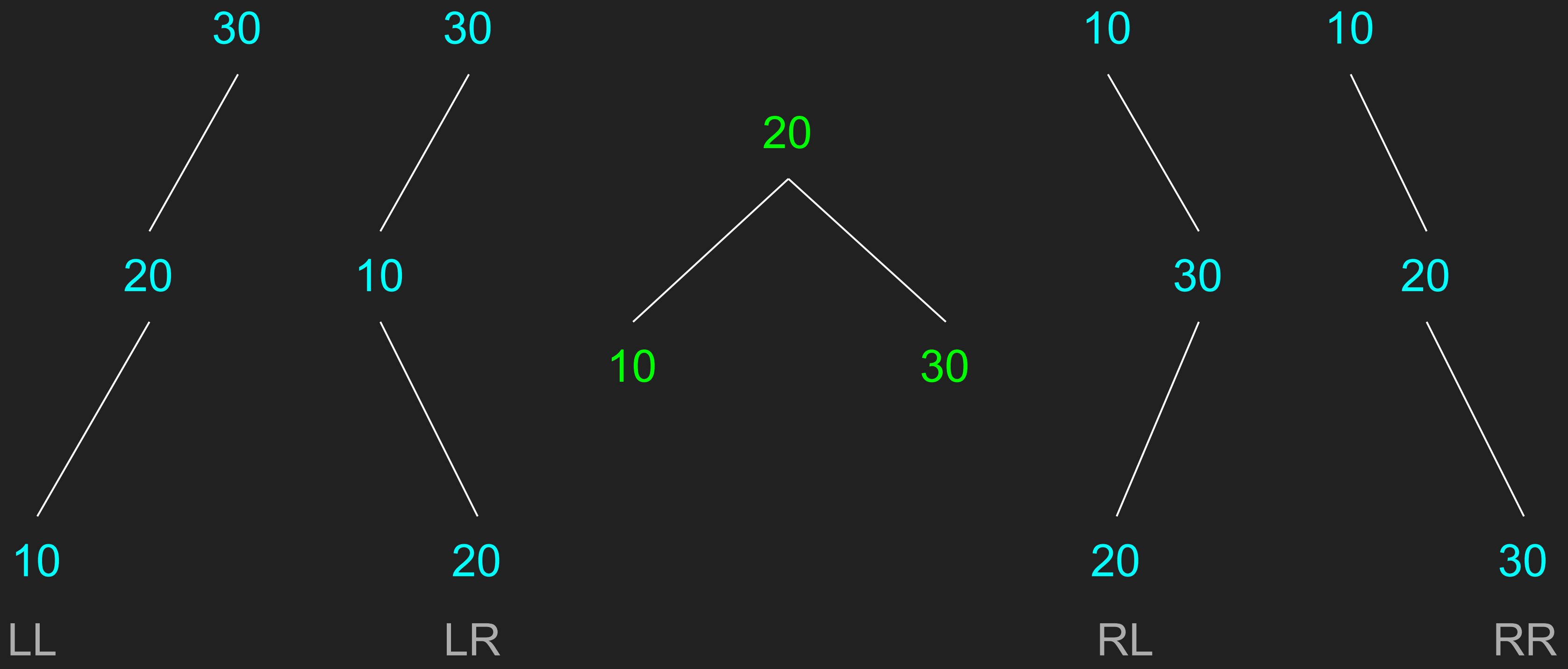
Árvores AVL: inserção

Exemplos…



Árvores AVL: inserção

Exemplos…



Árvores AVL: implementação rotação LL

```
No * rotacaoL(No * p){
    No * v;
    No * u = p->esq;
    if(u->bal == -1) { // rotação LL
        p->esq = u->dir;
        u->dir = p;
        p->bal = u->bal = 0;
        return u;
    } ...
```

Árvores AVL: implementação rotação LR

```
...
else if (u->bal == 1) { // LR
    v = u->dir;
    u->dir = v->esq;
    v->esq = u;
    p->esq = v->dir;
    v->dir = p;
    ...
```

Árvores AVL: implementação rotação LR

```
        if(v->bal == -1) p->bal = 1;
        else p->bal = 0;
        if(v->bal == 1) u->bal = -1;
        else u->bal = 0;
        v->bal = 0;
        return v;
    }
}
```