Aula dia 07/02/22 - Revisão de A.B.B.

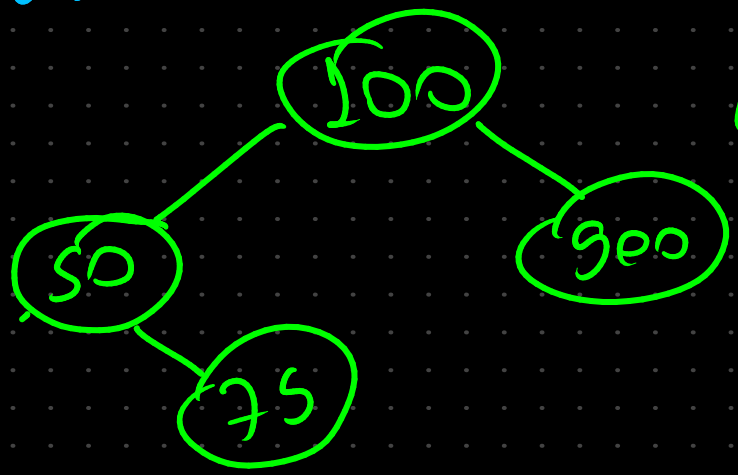
Definição. Uma árvore binária de busca é uma estrutura com as seguintes características:

1- É vazia, ou;

2- Possui um nó chamado raiz e 2 filhos, chamados esquerdo e direito, que são árvores binárias de busca.

3 Para todo nó de uma A.B.B. a chave desse nó é maior que toda chave na subárvore esquerda e é menor que toda chave na subárvore direita

Ex de ABB



Busca na árvore binária de busca

0 - árvore vazia

1, e o ponteiro para o nó - elemento encontrado

2, 3 > e o ponteiro para o pai do elemento buscado de ele for inserido < esq dir

A inserção ela se utiliza da busca para saber onde o elemento será inserido

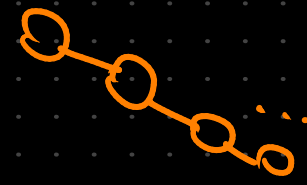
A remoção é sempre mais complicada.

Se o elemento for uma folha (nó sem filhos), basta remover e sinalizar ao elemento pai

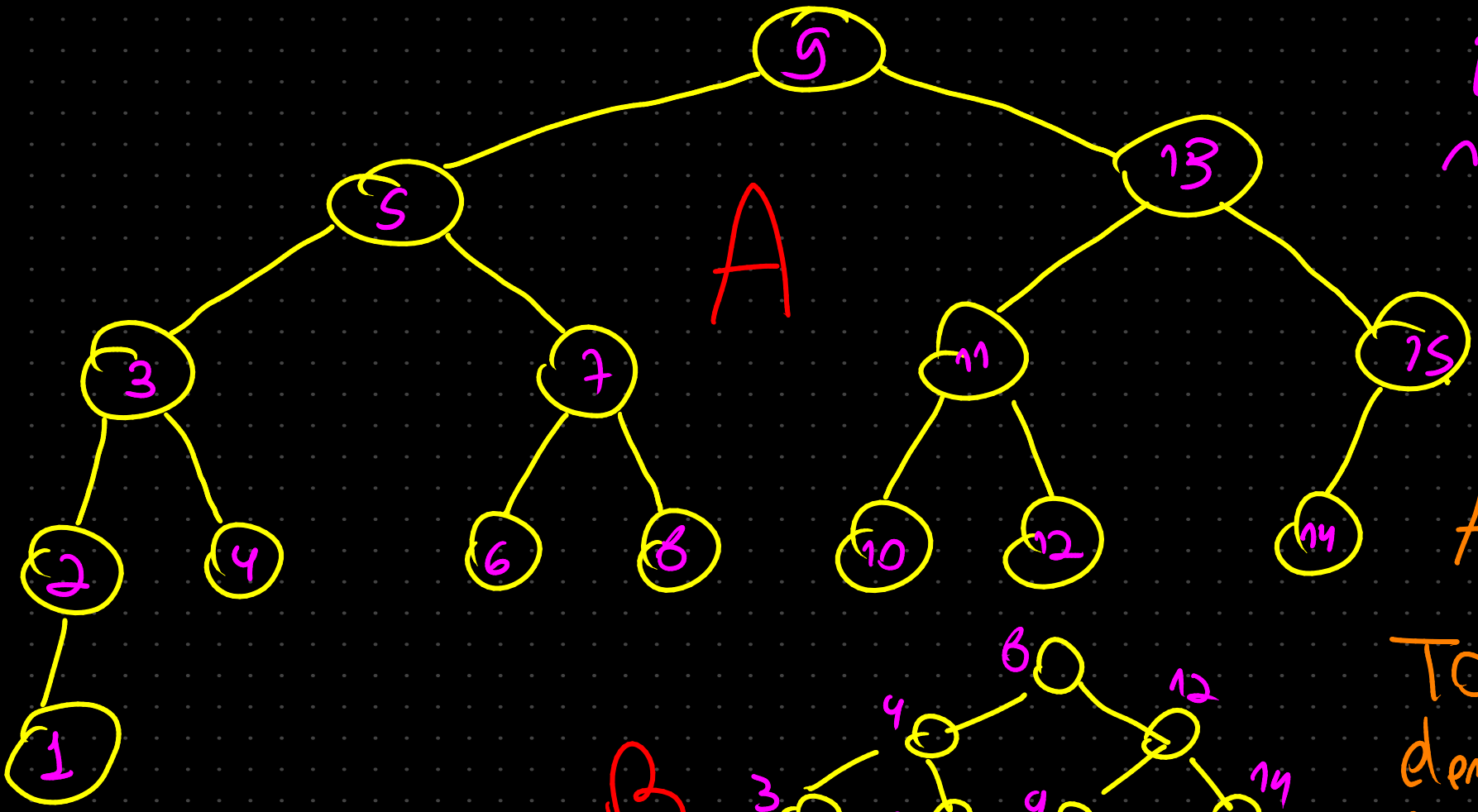
Se não for, substituimos o elemento pelo menor elemento maior que a chave e removemos esse elemento.

As complexidades	Melhor caso	Caso Médio	Pior Caso
Busca	$O(1)$	$O(\log n)$	$O(n)$ árvore zigzag
Inserção	Busca + $O(1)$	$O(1)$	$O(n)$
Remoção	Busca + $O(1)$	$O(\log n)$	$\Omega(n)$

O pior caso está sempre relacionado à forma da árvore. A forma da árvore depende da ordem de inserção dos elementos. Geralmente, não temos controle dessa ordem. Existem ordens que geram árvores boas e árvores ruins

Um exemplo se dá quando a lista de entrada está ordenada  árvore degenerada em lista

Será que é possível corrigir a minha árvore para que ela fique sempre uma árvore completa



A

Essa árvore não é completa

Para transformar A em B

B

Todos os elementos trocam de lugar

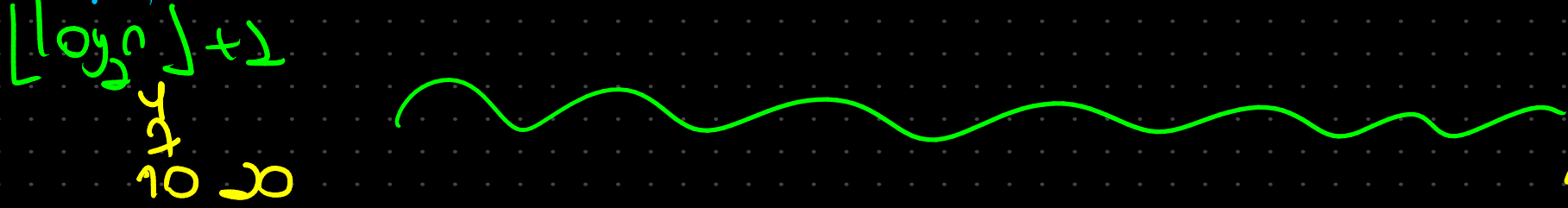
$\Omega(n)$

A abordagem de "consertar" a árvore para que ela esteja sempre completa é $\Omega(n)$. Um custo muito caro para ter benefício de uma busca $O(\log n)$

Lembremos que a altura de uma A.B.B completa é sempre $\lfloor \log_2 n \rfloor + 1$. É essa característica que nos dá a busca $O(\log n)$. Porém $\lfloor \log_2 n \rfloor + 1$ é muito forte. Se a altura for $O(\log n)$, já garantimos a busca $O(\log n)$ que é o que queremos.

A altura da árvore varia de

$\lfloor \log_2 n \rfloor + 1$



Será que existe algo a mais que garanta que a altura de uma A.B.B. com n elementos possua altura $O(\log n)$? SIM

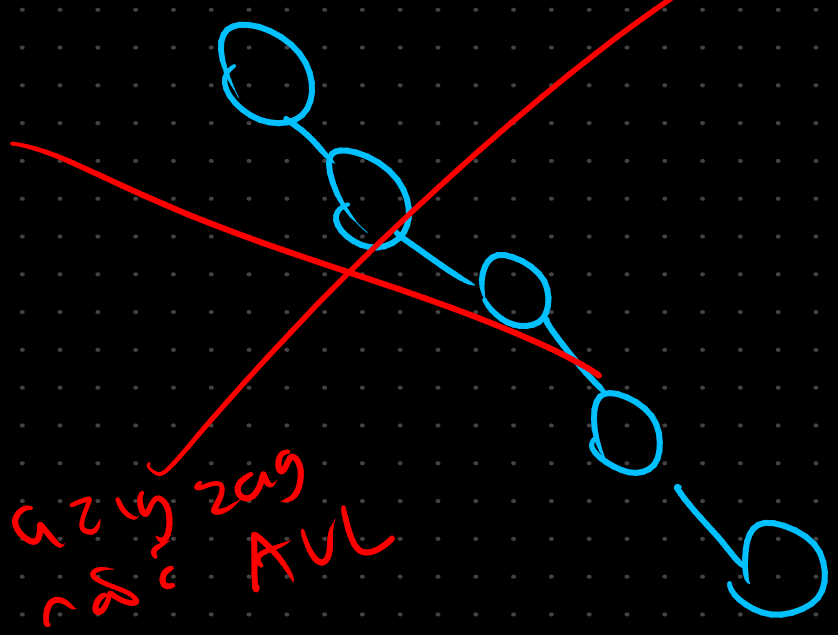
Definição: Uma árvore binária de busca balanceada é uma A.B.B. com altura garantida $O(\log n)$.

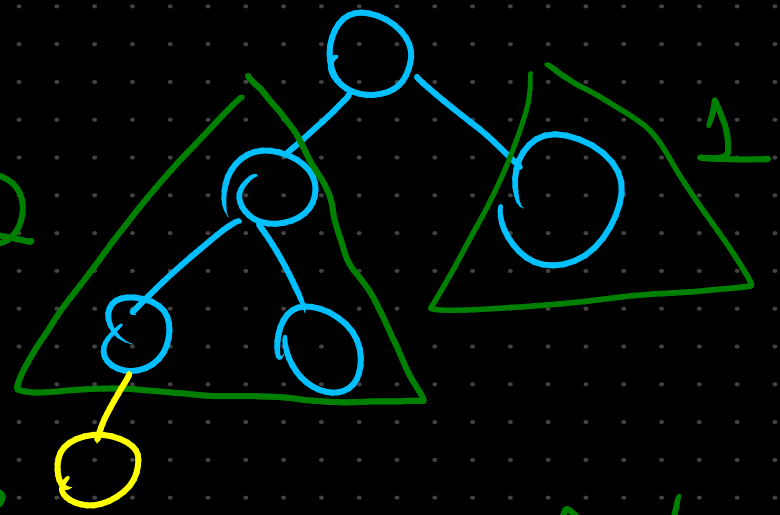
Na literatura, somos apresentados a dois tipos principais:

AVL Rubro-Negra

Essas A.B.B. possuem propriedades extras que devem ser sempre respeitadas na inserção e remoção. A busca é mesma

Uma A.B.B. é AVL se para todo nó da árvore a diferença de altura entre os filhos esquerdo e direito é no máximo 1.

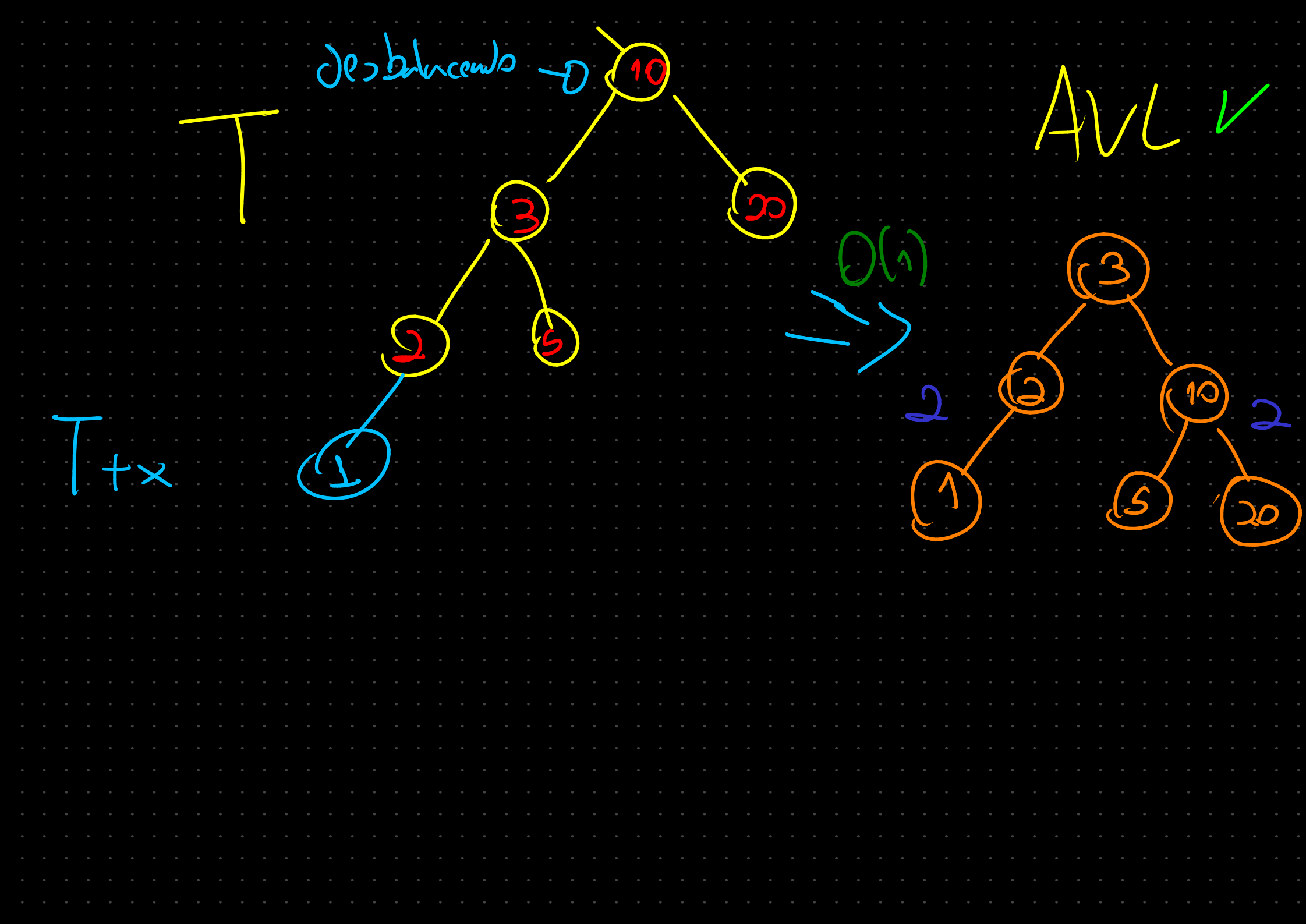




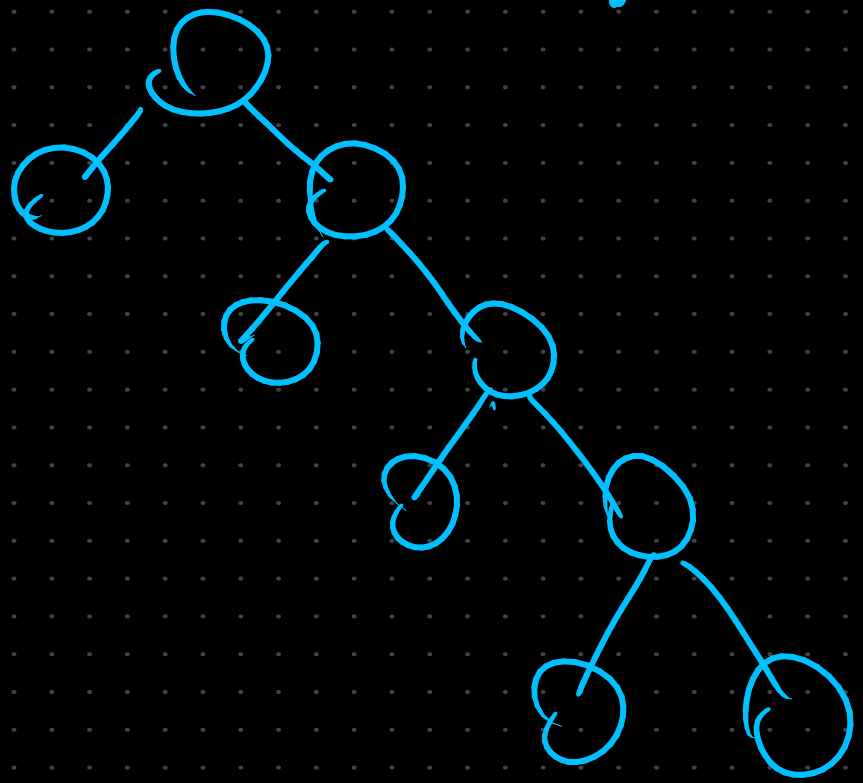
Árvores Cheia e Completa são AVL

Seja T uma árvore A.V.L.

Se ao inserirmos um elemento x em T e T+x não for AVL. O custo de transformar T+x em uma AVL é $O(1)$


Desbalanceado
T
10
3
20
2
5
1
T+x
O(1)
AVL
3
2
10
2
1
5
20

Se, algum dia, precisarmos implementar uma árvore binária de busca e o fizermos sem balancear a árvore, corremos o risco de não ter ganho de eficiência!



Altura $\lfloor n/2 \rfloor + 1$

$n/2+1 \Rightarrow O(n)$