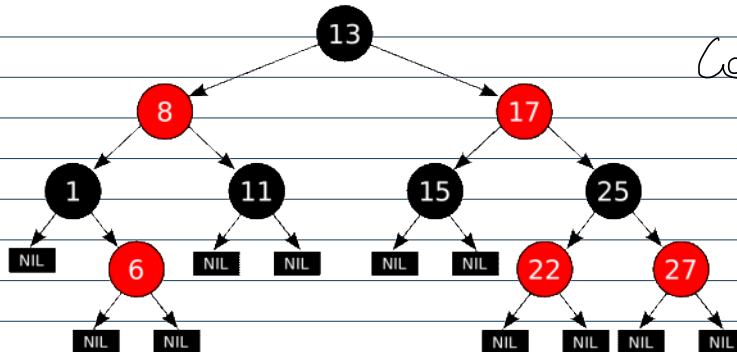


É um tipo de árvore binária de busca em que cada nó:

- tem um valor,
- tem uma cor (vermelha ou preta),
- ponteiros para SE, SD e pai (caso seja usado duplo encodamento).



Complexidade: $O(\log n)$

↳ + rápida na inserção e remoção que a AVL (exige - rotações)

↳ AVL tem busca + rápida por ser + balanceada.

* Propriedades

- todo nó da árvore é vermelho ou preto
- raiz é preta
- folhas ("nil") são pretas
- nós vermelhos tem ambos os filhos pretos
- p/ todo nó, o caminho do nó até as folhas contém o mesmo nº de nós pretos

Todos os nós internos contém chaves e os folhas são "nil"

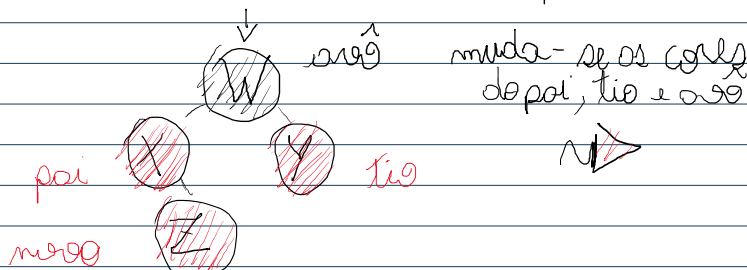
Geralmente, na implementação, usamos um nó sentinela de cor preta "NIL" p/ o qual todas as folhas apontam. Caso não o usemos, precisamos considerar nós nulos como pretos.

* Inserção: p/ garantir o balanceamento, todas as inserções (exceto a da raiz) são feitas p/ nós vermelhos.

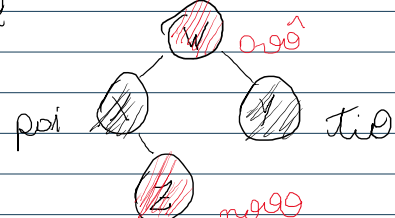
↳ se o pai for preto, não há problemas, mas, caso contrário, precisamos ajustar o nó pai, tio e avô.

→ Caso 1: pai/tio vermelhos e avô preto.

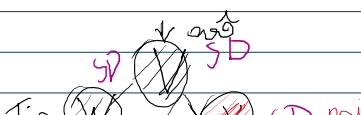
caso seja a raiz, deve ficar preto.



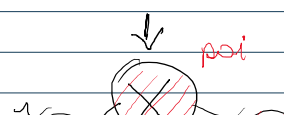
↳ novos nós são sempre vermelhos.

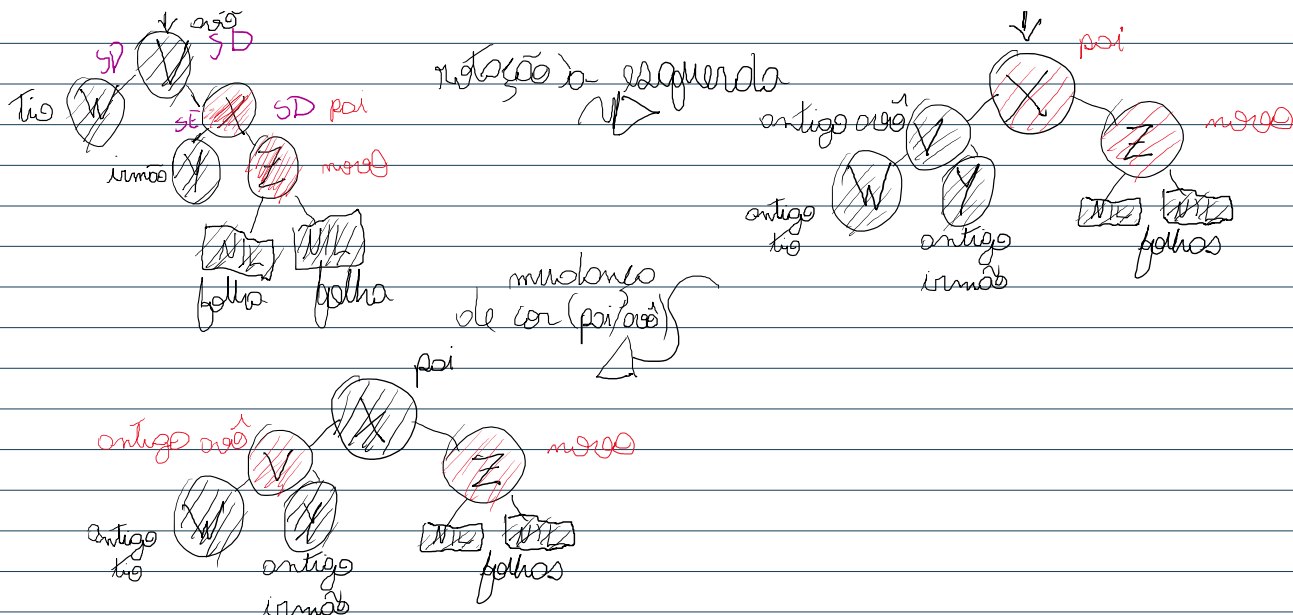


→ Caso 2: pai vermelho, tio preto, avô preto e caso esteja sendo inserido à direita do pai e o pai esteja à direita do avô.

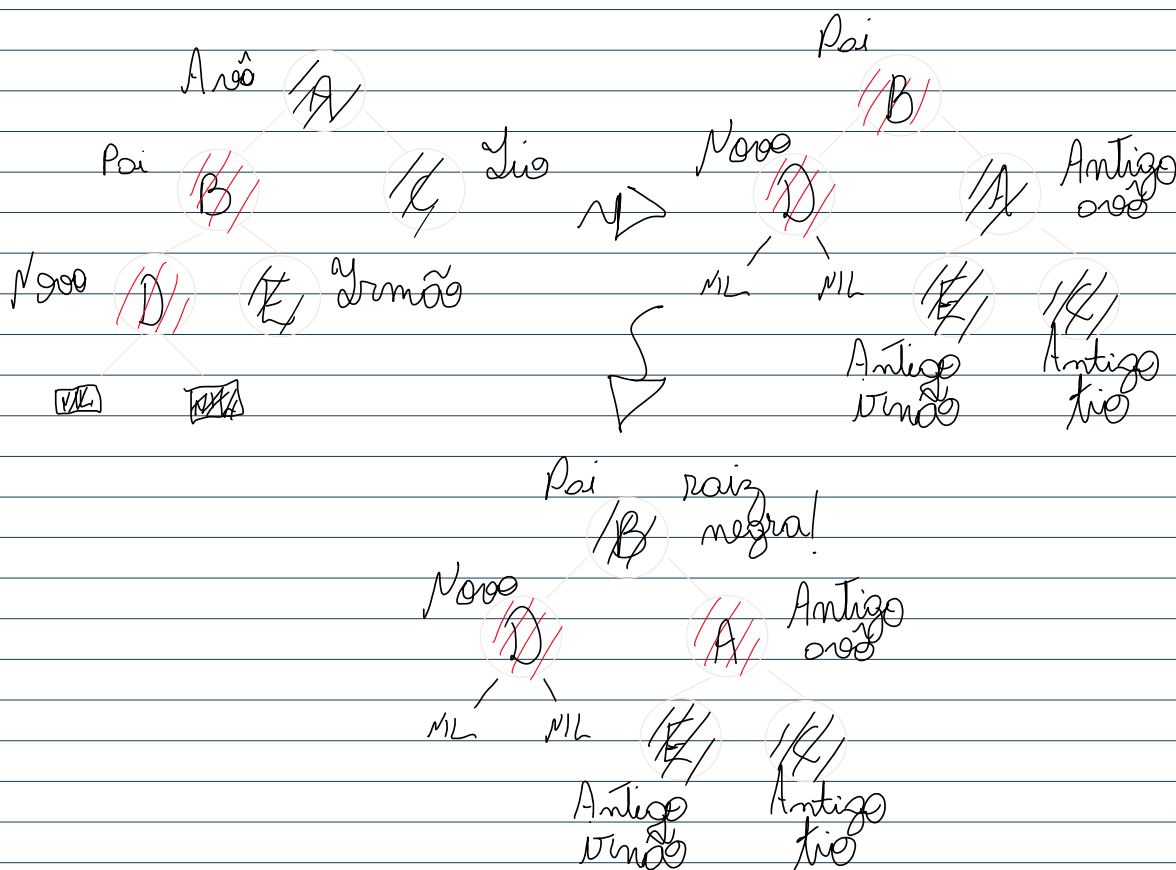


rotação à esquerda





⇒ Caso 3: pai é vermelho, tio preto, avô preto e o elemento foi inserido à esquerda do pai (que está à esquerda do avô).



⇒ Caso 4: pai é vermelho, tio preto, avô preto e o elemento foi inserido à esquerda do pai (que está à direita do avô).

