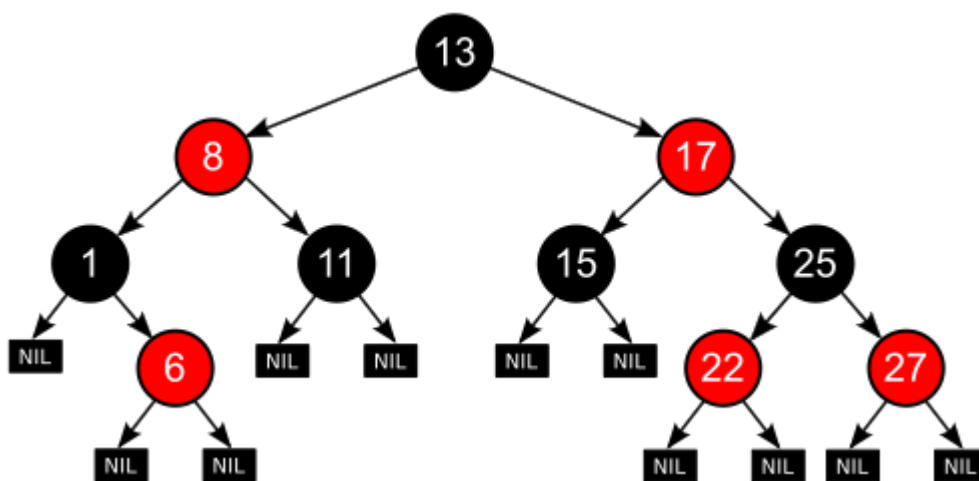


# Árvore Rubro Negra

## Introdução

A árvore Rubro negra é um tipo de árvore binária balanceada, originalmente criada por Rudolf Bayer em 1972 e chamada de Árvores Binárias Simétricas. Adquiriu o nome de Árvore Rubro-Negra somente após um trabalho de Leonidas J. Guibas e Robert Sedgwick em 1978.

A Árvore Rubro-Negra utiliza um esquema de coloração, diferenciando-se da Árvore AVL que utiliza um sistema de altura.



## Características Principais

- Nó da árvore possui um atributo de cor que pode ser “**vermelho**” ou “**preto**”
- A **raiz** é sempre “**preta**”
- Todo **nó folha** (“**NULL**”) é **preto**, como pode ser visto acima.
- Se um nó é **vermelho**, então os seus filhos são **pretos**, ou seja, **não existem nos vermelhos consecutivos**.
- Para cada nó, **todos** os caminhos desse nó para os nós folhas descendentes contém o mesmo número de nós “**pretos**”
- Como todo nó folha termina com dois ponteiros para “**NULL**”, eles podem ser ignorados na representação da árvore para fins de didática;

## Funcionamento

A Árvore **Rubro-Negro** permite **rebalanceamento local**, ou seja, apenas a parte afetada pela inserção ou remoção é rebalanceada e para a realização do rebalanceamento, conta com o ajuste de cores e rotações, corrigindo propriedade violadas.

A árvore rubro-negra busca manter-se como uma **árvore binária quase completa**. O custo de qualquer algoritmo é máximo " $O(\log N)$ ", sendo **N** o número de nós, a expressão se refere ao custo de operações de busca, inserção e remoção em uma árvore rubro-negra. O " $O(\log N)$ " indica que o tempo de execução dessas operações é limitado superiormente por um fator logarítmico do número de nós na árvore. Isso é altamente eficiente, especialmente quando comparado com árvores binárias de busca não balanceadas, onde o pior caso pode ser uma árvore degenerada, levando a um tempo de execução de  $O(N)$  para essas operações.

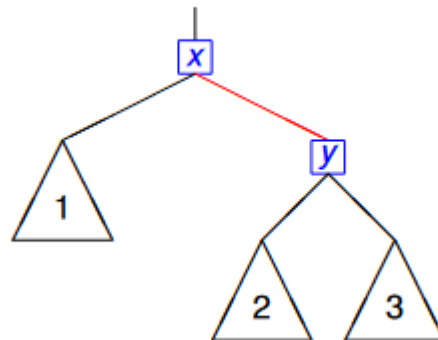
A altura  $h$  de uma árvore rubro-negra de  $n$  chaves ou nós internos é no máximo  $2 \log(n + 1)$ .

Inserções e remoções feitas numa árvore rubro-negra pode modificar a sua estrutura. Precisamos garantir que nenhuma das propriedades de árvore rubro-negra seja violada.

Para isso podemos ter que mudar a estrutura da árvore e as cores de alguns dos nós da árvore. A mudança da estrutura da árvore é feita por dois tipos de rotações em ramos da árvore: Left Rotate e Right Rotate.

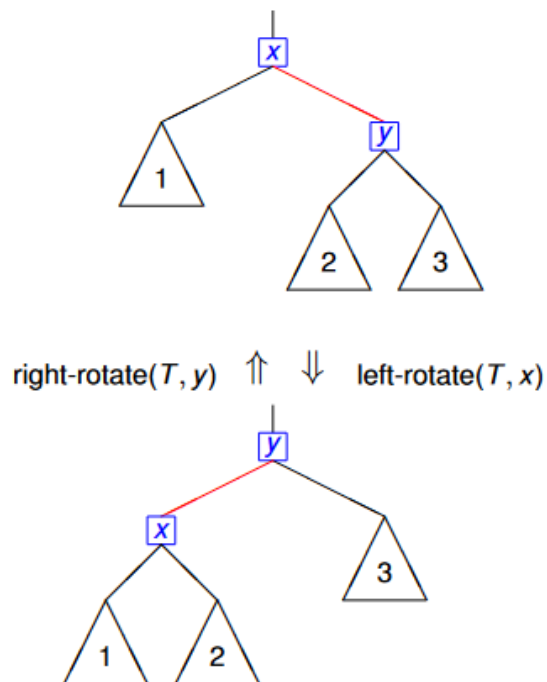
---

Seja uma árvore binária apontada por  $T$



---

Seja uma árvore binária apontada por  $T$



## Árvore Rubro-Negra vs AVL

Na teoria, as duas possuem a mesma complexidade computacional, realizando inserção, remoção e busca. Na prática, a árvore AVL, por ser mais balanceada que a árvore rubro-negra, é mais rápida na operação de busca e mais lenta nas operações de inserção e remoção. Mas apresenta um maior custo de rotações nos nós, o que torna as outras operações mais lentas.

A escolha pela aplicação de Árvore Rubro-Negra ou Árvore AVL em sua aplicação, deve ser baseada em sua necessidade. Caso sua aplicação tenha buscas mais intensas e constantes, é mais interessante utilizar a árvore AVL, caso contrário, por ser mais rápida nas operações de inserção e remoção, a Árvore Rubro-Negra passa a ser a mais comum e mais indicada no uso geral.

Um exemplo interessante é quando lidamos com Bancos de dados com alto tráfego de inserção/remoção e leitura. Em sistemas de banco de dados, é comum precisar inserir e remover dados continuamente. Se você estiver usando uma árvore para indexar esses dados, uma árvore rubro-negra pode ser preferível, pois é mais flexível em relação ao balanceamento. Isso significa que **a árvore rubro-negra pode acomodar inserções e remoções frequentes sem precisar reequilibrar a árvore com tanta frequência** quanto uma árvore AVL. Enquanto **uma árvore AVL manterá um equilíbrio rígido o tempo todo, o que pode ser caro em termos de tempo de CPU** durante operações de inserção/remoção, uma árvore rubro-negra pode permitir uma ligeira desigualdade entre as

alturas das subárvores esquerda e direita, o que pode resultar em inserções/remoções mais eficientes.