

Assim como a árvore AVL, a rubro-negra, ou vermelho-preto, também são árvores de pesquisa binária, só que possui um bit extra no armazenamento do nó, a sua cor. A cor do nó pode ser somente vermelho ou preto, dessa forma, restringe o modo de como os nós são coloridos em qualquer caminho da árvore. A árvore rubro-negra garante que nenhum dos caminhos será maior que duas vezes o comprimento de qualquer outro caminho, sendo assim, é considerada uma árvore aproximadamente balanceada.

Essa árvore possui cinco propriedades:

1. Todo nó é vermelho ou preto;
2. A raiz é preta;
3. Toda folha é preta;
4. Se um nó é vermelho, então ambos os filhos são pretos;
5. Para cada nó, todos os caminhos de um nó até as folhas descendentes contêm o mesmo número de nós.

A altura de preto de uma árvore rubro-negra é o caminho de nós pretos em qualquer caminho desde um determinado nó, não incluindo esse nó, até um nó folha. Podemos definir a altura de preto de uma rubro-negra é a altura de preto da raiz da árvore.

As operações de inserção e remoção em uma árvore possuem complexidade de $O(\log n)$. Essas operações, quando executadas, podem modificar a árvore, assim, violando uma das cinco propriedades citadas anteriormente. Para corrigir isso, são utilizadas as rotações, que podem ser à direita ou à esquerda e que funcionam da mesma maneira que nas árvores AVL.

Na inserção, quando há quebra de alguma propriedade da árvore rubro-negra, existem três casos (três do lado esquerdo e três do direito, totalizando 6 casos) que são usados para corrigir as cores da árvore. O primeiro caso é quando o tio do nó avaliado é da cor vermelha. O segundo é quando o tio do nó avaliado possui a cor preta e o nó avaliado é o filho da direita. Por fim, o terceiro caso é quando o tio do nó avaliado possui a cor preta e o nó avaliado é o filho da esquerda.

Quando ocorre o primeiro caso, o que se faz é mudar a cor a do pai e do tio do nó para preto e a cor do avô do nó para vermelho, depois disso deve-se avaliar recursivamente o avô do nó, para saber em qual dos casos ele se encaixa. No segundo caso, deve-se fazer uma rotação à esquerda no do pai (e que passa a ser avaliado) e isso levará ao terceiro caso. No terceiro caso, muda a cor do pai do nó para preto, a cor do avô do nó para vermelho e faz uma rotação à direita do avô. Os passos são o mesmo caso o nó avaliado esteja do lado direito do avô, porém devem ser espelhados, trocando a esquerda e a direita.

A processo de remoção é um pouco mais complexo que o de inserção. A remoção de um nó vermelho não quebra as propriedades da árvore rubro-negra, diferentemente da remoção de um nó preto. Sendo assim, o processo de restauração das propriedades da árvore só precisará ser feito se o nó removido tiver a cor preta. O nó que será usado para atualizar o balanceamento da árvore depois da remoção é o que passou a ocupar a posição do nó removido da árvore.

Existem 3 situações que podem acontecer: a primeira é se o nó removido for vermelho, o nó removido for preto e ter sucessor vermelho ou o nó removido for preto e ter sucessor preto. Na primeira situação, a remoção do nó vermelho não irá alterar a altura de preto da árvore, portanto não será necessário fazer nada. Na segunda situação, a remoção do nó preto irá alterar em uma unidade a altura de preto da árvore. Com isso, deve-se colorir de preto o nó sucessor (antes era vermelho), já que ele assumiu o lugar do nó removido. A terceira situação é um pouco mais complexa, necessitando utilizar o irmão do nó a ser removido. Nessa situação podem ocorrer quatro casos: se o irmão do nó for vermelho, se o irmão é preto e ambos os seus filhos também são pretos, o irmão é preto e o seu filho a esquerda é vermelho ou o irmão é preto e o seu filho a direita é vermelho.

No primeiro caso, deve-se colorir o irmão de preto, colorir o pai do nó de vermelho e depois fazer uma rotação à esquerda no pai, levando ao caso 2, 3 ou 4. No segundo caso, deve-se colorir o irmão de vermelho e irá andar na árvore, recursivamente, até que o ponteiro que aponta pro nó removido seja a raiz ou a cor de x for vermelho). No terceiro caso, deve-se colorir de preto o filho a esquerda do irmão, colorir o irmão de vermelho e fazer rotação à direita no irmão, levando ao caso 4. No último caso, o quarto, deve-se colorir o irmão com a mesma cor do pai, colorir o pai de preto, colorir o filho a direita do irmão de preto e fazer uma rotação à esquerda no pai do nó removido. No final, o ponteiro que apontava pro nó removido deverá apontar para a raiz da árvore.

O processo será o mesmo caso o nó a ser removido esteja a direita do pai, porém o algoritmo deverá ser espelhado, ou seja, substituir direita por esquerda e esquerda por direita.