RESUMO ED2

ÁRVORE RUBRO-NEGRA

Rubro-Negra

Uma árvore rubro-negra é uma árvore de busca binária autobalanceável que possui um bit extra de armazenamento, sua cor. Essas árvores asseguram que o comprimento de nenhum desses caminhos seja maior que duas vezes o de qualquer outro, de modo que a árvore seja aproximadamente balanceada. No caso dessa árvore trataremos o nó que aponta para o pai, caso seja nulo, como ponteiros para as folhas, nós externos, dela e os nós normais que possuem chaves como nós internos da árvore.

A árvore rubro-negra deve suprir as seguintes características:

1. Todo nó deverá ser vermelho ou preto
2. Se um nó é vermelho, seus filhos são pretos
3. Toda folha é preta
4. A raiz é preta
5. Todos os caminhos simples de um nó até as folhas contém o mesmo número de nós pretos

Propriedades Gerais

Cor do nó: cada nó é vermelho ou preto

Folhas: Todas as folhas são representadas pelo ponteiro NIL na árvore e são pretas.

Raiz: O nó raiz é preto

Altura das folhas pretas: Todos os caminhos simples de um nó até as folhas contém o mesmo número de nós pretos

Balanceamento: A altura da árvore será no máximo 2log (n + 1), onde n é o número de nós internos da árvore.

Tempo de busca: O tempo de busca na árvore rubro-negra é de O(log n)

Inserir e Deletar: O tempo de inserção e deletar da árvore é O(log n), porém diferente da árvore de busca binária a árvore-rubro-negra é autobalanceavel garantindo que a altura dela será no máximo 2log n – 1, e por conta do autobalanceamento a complexidade fica em torno de O(log n). O máximo leva em consideração o pior caso onde metade dos nós da árvore estão coloridos em vermelho, o que pode dobrar a altura dela.

Rotações

As operações de árvore de busca na árvore rubro-negra possuem complexidade de O(log n) e podem violar as propriedades da árvore rubro-negra. Por isso, utilizamos uma técnica para mudar a estrutura de ponteiros chamada de rotação. Temos dois tipos de rotação, para a esquerda e para a direita. A rotação para a esquerda ocorre quando a subárvore da direita é mais alta que a subárvore da esquerda. Essa operação consiste em rotacionar o nó pai e seu filho da direita para a esquerda fazendo com que o nó pai e o nó da esquerda se torne filho do nó da direita. Essa operação resulta em uma subárvore esquerda mais alta. Já a rotação para a direita envolve rotacionar o nó e seu filho da esquerda para a direita, fazendo com que o filho da direita se torne pai do nó original e de seu filho da direita, essa operação ocorre quando a subárvore esquerda é maior que a subárvore direita.

As rotações são utilizadas para balancear a árvore e manter a altura dela como O(log n).