Teoria das Árvores

Árvores Binárias

- **Definição:**

Árvores binárias de pesquisa que são projetadas para um acesso rápido à informação. Idealmente a árvore deve ser razoavelmente equilibrada e a sua altura será dada (no caso de estar completa) por h=log2 (n+1) O tempo de pesquisa tende a O(log2N) Porém, com sucessivas inserções de dados principalmente ordenados, ela pode se degenerar para O(n).

- **Árvores Completas:**

São aquelas que minimizam o número de comparações efetuadas no pior caso para uma busca com chaves de probabilidades de ocorrências idênticas.

Inserção de elementos em uma árvore binária de busca

• O primeiro elemento inserido assumirá o papel de raiz da árvore;
• Todo novo elemento entrará na árvore como uma folha;
• Se o elemento for menor ou igual à raiz será inserido no ramo da esquerda. Caso contrário, no ramo da direita (para árvores decrescentes inverte-se a regra).

Remoção de elementos em uma árvore binária de busca

Considerando que podemos remover qualquer elemento de uma árvore, podem ocorrer as seguintes situações:

1. O Elemento a ser removido é um nó folha (sem filhos à esquerda e à direita);
2. O Elemento a ser removido possui apenas um filho (à direita ou à esquerda);
3. O Elemento a ser removido possui dois filhos.

- ## Travessia da árvore:

- **pré-ordem** (os filhos de um nó são processados após o nó)

    - Cima para baixo
- **pós-ordem** (os filhos são processados antes do nó)
    - Baixo para cima
- **em-ordem** , em que se processa o filho à esquerda, o nó, e finalmente o filho à direita.

- Métodos de Balanceamento

- Há duas categorias: dinâmico e global (ou estático).
- O rebalanceamento dinâmico mantém a árvore balanceada toda vez que um nó é inserido ou removido. AVL é o melhor exemplo
- O global permite a árvore crescer sem limites e somente faz o balanceamento quando tal necessidade é acionada, externamente.

Existe alguma razão para evitarmos algoritmos de busca em árvore recursivos (memória, complexidade, etc)? Justifique.

Sim, pois quando essa busca acontece no pior dos casos, ou seja, a árvore é degenerada, a busca irá percorrer todos os nós de um lado, apenas para depois desempilhar os valores na pilha de recursão e retornar nulo quando chegar na raiz, caso não encontre o valor. Isso faz com que a complexidade do algoritmo aumente, sendo ele O(logn), consequentemente aumentando seu tempo de execução.

E ainda existem outras formas de busca, como a busca iterativa, que também possui loops mas que não utilizam recursão, e sim repetição, resultando numa complexidade e tempo de execução menores.