**ÁRVORES BINÁRIAS**

Realize uma leitura dos materiais indicados e faça um resumo de 3 páginas explicando o seu entendimento sobre árvores, contendo (no mínimo):

1. Definição de árvore, folha, nó ancestral, nó descendente

1.1.0 Definição de árvore

Estrutura de dados não lineares muito utilizada para representação de dados hierárquicos, as árvores são compostas por um conjunto de nós. Cada árvore possui um nó raiz, com 0 ou mais subárvores a direita e a esquerda que por sua vez tem também 0 ou mais subárvores a esquerda ou à direita e poderíamos repetir isso quantas vezes tivéssemos tempo nos passando a ideia de recursividade infinita, os nós raízes da subárvores são como filhas do nó raiz, que são ligadas diretamente ao nó raiz ou nó pai. Um nó pai com filhos é chamado de nó interno e nós sem filhos são chamados de folhas ou externos. No mundo computacional é normal as árvores serem ao contrário das árvores tradicionais que vemos na natureza, com a raiz para cima e as folhas para baixo, o número de filhos e o tipo de informações armazenadas em cada nó da árvore diferenciam os tipos de árvores.

1.1.1 Folha

Um nó sem filhos é chamado de nó folha, não tem descendentes, mas cada nó folha representa subárvores vazias direita e esquerda.

1.1.2 Nó ancestral e descendente.

Um nó raiz será o ancestral e está acima de outro nó e seus filhos ou folhas, e possuem ligação direta ou indireta. Já o nó filho é um descendente e está abaixo do nó raiz e também possuem ligação direta e indireta, os nós a direita e esquerda são nós irmãos.

2. Exemplifique uma árvore estritamente binária

Em uma árvore estritamente binária todos os nós possuem 0 ou 2 filhos, todo nó que não for folha e tiver uma subárvore à direita e a esquerda não vazia será uma árvore estritamente binária. Uma árvore completa tem todos os nós com menos de dois filhos e estão no último e penúltimo nível uma árvore cheia é uma árvore estritamente binária, em que se um nó tem alguma sub-árvore vazia então ele está no último nível. Uma árvore completa é aquela em se ***n*** é um nó com alguma sub-árvores vazias, então ***n*** se localiza no penúltimo ou no último nível. Portanto, toda árvore cheia é completa e estritamente binária.

3. Explique, textualmente, como são realizadas operações em árvores binárias (incluir, remover, atualizar)

3.1.0 Incluir

Na operação de inserção as propriedades de uma árvore devem ser obedecidas e todo o novo nó é considerado uma folha, construímos uma função para criar a árvore vazia representada pelo valor NULL e podemos criar uma função com parâmetros que consiste de uma chave que será o nó raiz e ponteiros para a esquerda e para a direita que retorna o endereço do nó raiz criado, valores que são menores que a raiz serão inseridos na subárvore direita do nó raiz e valores menores na subárvore esquerda , essa função é recursiva e será chamada dentro dela mesma para inserimos quanto valores quisermos. Começamos do nó raiz da árvore e descemos de modo recursivo, procurando pelo local certo para inserir o novo nó. Se um nó com o mesmo valor já existir na árvore, é possível escolher inserir a duplicata ou não. Algumas árvores permitem duplicatas, outras não. Isso depende da implementação.

3.1.1 Remover

Podemos excluir um nó passando por três cenários diferentes:

* Apenas um nó folha, que é um nó que não possui filhos. Eles são facilmente removidos.
* Um nó com filho, que se o nó tiver um filho na parte esquerda, atualizaremos o ponteiro (referência) do nó pai para o filho esquerdo do nó a ser excluído. O mesmo acontece com o filho da parte direita.
* Um nó com dois filhos, onde encontramos o nó com valor mínimo na parte direita e substituímos esse nó por o nó com valor mínimo e removemos da árvore.

3.1.2 Atualizar

Para atualização de uma árvore utilizamos operações com algoritmos de rotação na árvore que pode ser simples ou duplo tanto para a direita como para a esquerda de acordo com a necessidade de balanceamento da árvore. Há também algoritmos que atualizam automaticamente quando um novo nó é inserido ou deletado, em algumas linguagens e atualizado sem necessidade de implementar, com garbage coletor.

4. análise de complexidade de árvores

A relação existente entre altura (h) e número de nós (n) em árvores binárias é uma importante informação de muitas aplicações, possui altura máxima aquela que cada nó possui apenas um único filho. A altura de tal árvore é igual a n, uma árvore completa tem uma altura mínima. A complexidade de busca depende da altura da árvore 0 (log n), na operação de remoção o pior caso e quando o dado a ser removido está em uma folha no nível mais baixo.

5. Faça um algoritmo que:

a. o número de nós numa árvore binária;

b. a soma do conteúdo de todos os nós numa árvore binária;

c. a profundidade de uma árvore binária.