

coggle
made for free at coggle.it

Procurar por um elemento V é feito recursivamente da seguinte maneira, se a árvore está vazia, a procura resulta em falha, se não está vazia nós comparamos o elemento com a raiz $K(r)$ da árvore

Se eles são iguais, a busca é finalizada, senão, continua-se a busca na sub árvore esquerda da raiz, se $V < K(r)$ e na sub árvore direita se $V > K(r)$

O pior caso de eficiência da operação de busca é $\Theta(n)$. Já, o caso médio está em $\Theta(\log n)$.

Procura

Inserção

Já que a inserção de uma nova chave em um árvore binária de busca é quase idêntica a busca, ela também exemplifica a técnica do decréscimo do tamanho da variável e tem a mesmas características de eficiência que a operação de busca

Procura e Inserção

Árvores

Uma árvore é um grafo acíclico conectado. Um grafo que não tem ciclos mas não é necessariamente conectado é chamado de floresta, cada um de seus componentes conectados é uma árvore. O número de arestas de uma árvore sempre é um a menos que o número de vértices

Transversais e Propriedades da Árvore Binária de Busca

Uma árvore binária T é definida como um set finito de nós que ou está vazia, ou consiste de uma raiz e duas árvores binárias disjuntas TL e TR , chamadas, respectivamente de a sub árvore esquerda e direita da raiz.

Muitos problemas sobre árvores binárias podem ser resolvidas aplicando a técnica de dividir e conquistar

O mais importante algoritmo dividir e conquistar para árvores binárias são os três clássicos transversais: preorder, inorder, e postorder.

Na transversal preorder, a raiz é visitada antes que sub árvores esquerda e direita sejam visitadas (nessa ordem)

Na transversal inorder, a raiz é visitada após visitar sua sub árvore esquerda mas antes de visitar sub árvore direita

Na transversal postorder, a raiz é visitada após visitar as sub árvores esquerda e direita (nessa ordem)

