



Apache
NetBeans IDE



Prof.°

Alexandre Gomes

alexandre.silva251@fatec.sp.gov.br



XandyGomes



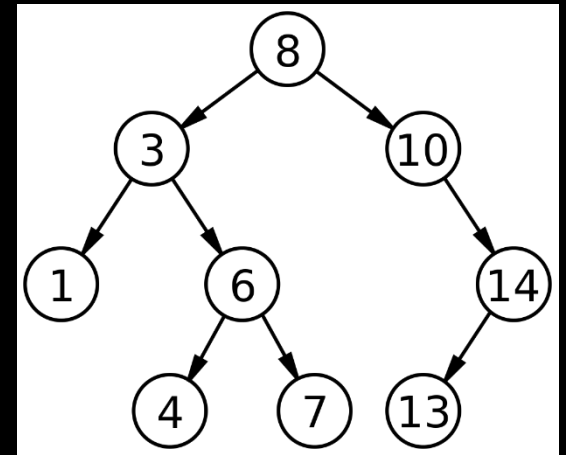
“ÁRVORE BINÁRIA DE BUSCA”

Uma árvore binária de busca (ou árvore binária de pesquisa) é uma estrutura de dados de árvore binária baseada em nós, onde todos os nós da sub-árvore esquerda possuem um valor numérico inferior ao nó raiz e todos os nós da sub-árvore direita possuem um valor superior ao nó raiz (esta é a forma padrão, podendo as sub-árvores serem invertidas, dependendo da aplicação).

O objetivo desta árvore é estruturar os dados de forma a permitir busca binária.

Uma árvore binária é uma estrutura de dados útil quando precisam ser tomadas decisões bidirecionais em cada ponto de um processo. Por exemplo, suponha que precisemos encontrar todas as repetições numa lista de números. Uma maneira de fazer isto é comparar cada número com todos que o precedem. Entretanto, isso envolve um grande número de comparações. O número de comparações pode ser reduzido usando-se uma árvore binária. O primeiro número na lista é colocado num nó estabelecido como a raiz de uma árvore binária com as sub-árvores esquerda e direita vazias.

Elementos



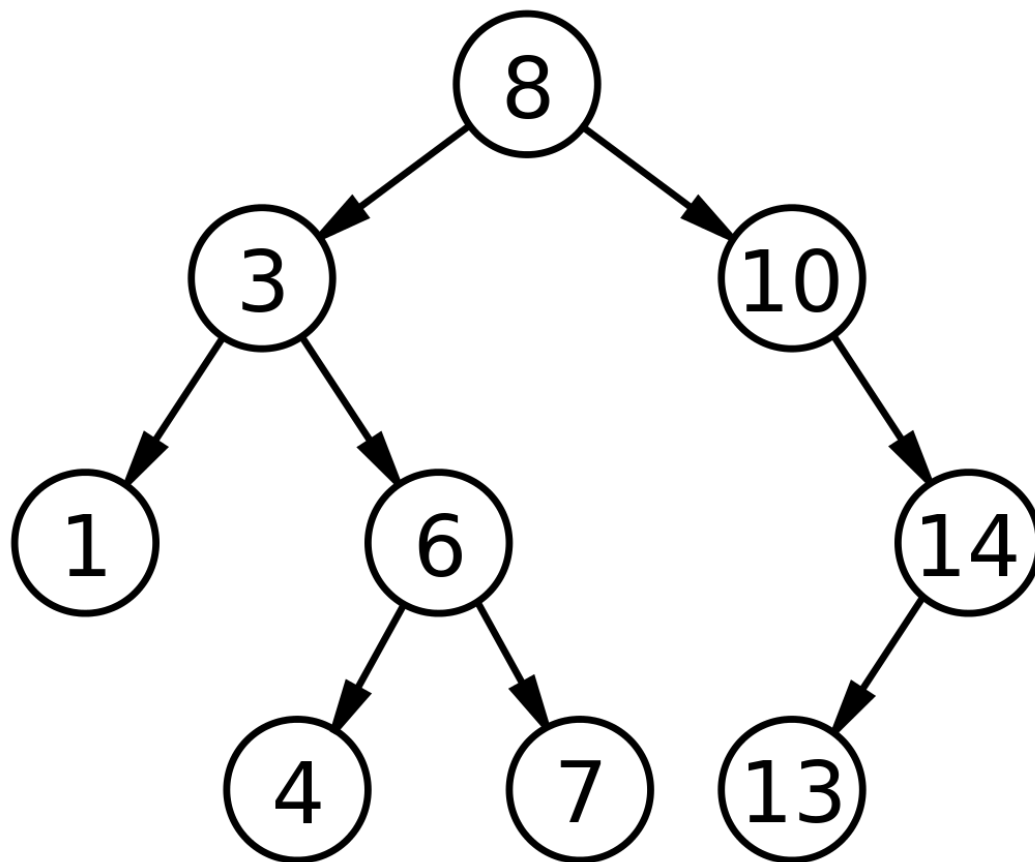
Nós - são todos os itens guardados na árvore

Raiz - é o nó do topo da árvore (no caso da figura acima, a raiz é o nó 8)

Filhos - são os nós que vem depois dos outros nós (no caso da figura acima, o nó 6 é filho do 3)

Pais - são os nós que vem antes dos outros nós (no caso da figura acima, o nó 10 é pai do 14)

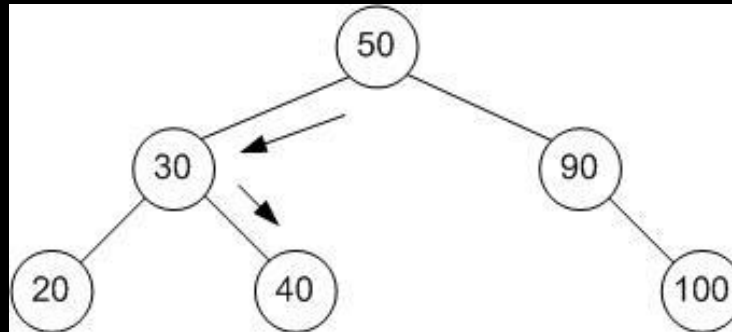
Folhas - são os nós que não têm filhos; são os últimos nós da árvore (no caso da figura acima, as folhas são 1, 4, 7 e 13)



Árvore binária de busca com a cardinalidade 9 e altura 4, com raiz 8 e folhas 1, 4, 7 e 13

Busca

A busca em uma árvore binária por um valor específico pode ser um processo recursivo ou iterativo. Será apresentado um método recursivo.



A busca começa examinando o nó raiz. Se a árvore está vazia, o valor procurado não pode existir na árvore. Caso contrário, se o valor é igual a raiz, a busca foi bem sucedida. Se o valor é menor do que a raiz, a busca segue pela subárvore esquerda. Similarmente, se o valor é maior do que a raiz, a busca segue pela subárvore direita. Esse processo é repetido até o valor ser encontrado ou a subárvore ser nula (vazia). Se o valor não for encontrado até a busca chegar na subárvore nula, então o valor não deve estar presente na árvore.

Pré-ordem ou Profundidade: (VLR)

Visita a raiz

Percorre a sub-árvore esquerda em pré-ordem

Percorre a sub-árvore direita em pré-ordem

In-ordem ou Ordem Simétrica: (LVR)

Percorre a sub-árvore esquerda em ordem simétrica

Visita a raiz

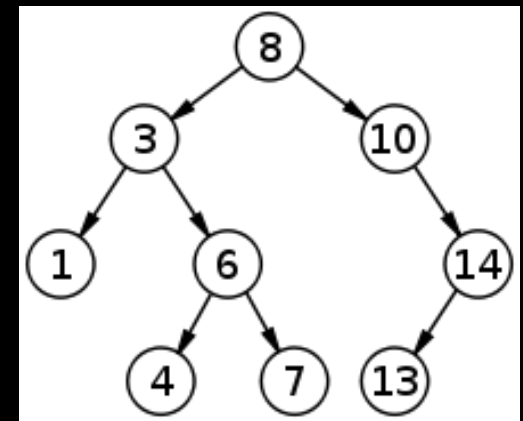
Percorre a sub-árvore direita em ordem simétrica

Pós-ordem: (LRV)

Percorre a sub-árvore esquerda em pós-ordem

Percorre a sub-árvore direita em pós-ordem

Visita a raiz



Pré-ordem => 8, 3, 1, 6, 4, 7, 10, 14, 13

Ordem simétrica => 1, 3, 4, 6, 7, 8, 10, 13, 14 (chaves ordenadas)

Pós-ordem => 1, 4, 7, 6, 3, 13, 14, 10, 8

Pré-ordem:

Aplicativo de Navegação de Arquivos: Quando você deseja mostrar a estrutura de diretórios de forma hierárquica, visitando primeiro a pasta raiz e, em seguida, listando seus subdiretórios antes de explorar os subdiretórios mais profundamente. Essa abordagem permite a visualização de toda a árvore de diretórios de forma que a raiz seja mostrada antes de seus conteúdos.

In-ordem (Ordem simétrica):

Software de Banco de Dados: Quando você deseja listar registros em uma ordem crescente com base em uma chave (por exemplo, listagem de produtos ordenados pelo preço). Uma árvore binária de busca pode ser usada internamente para garantir que os elementos sejam apresentados em ordem crescente.

Pós-ordem:

Software de Backup e Restauração: Se um sistema precisar fazer backup de arquivos e pastas, ele pode usar a travessia pós-ordem para garantir que todos os subdiretórios e arquivos sejam copiados antes da própria pasta pai. Isso garante que, ao restaurar, a estrutura da pasta possa ser reconstruída corretamente, de baixo para cima.