Fábio Henrique Alves Fernandes - 19.1.4128

Resumo: Pesquisa Externa

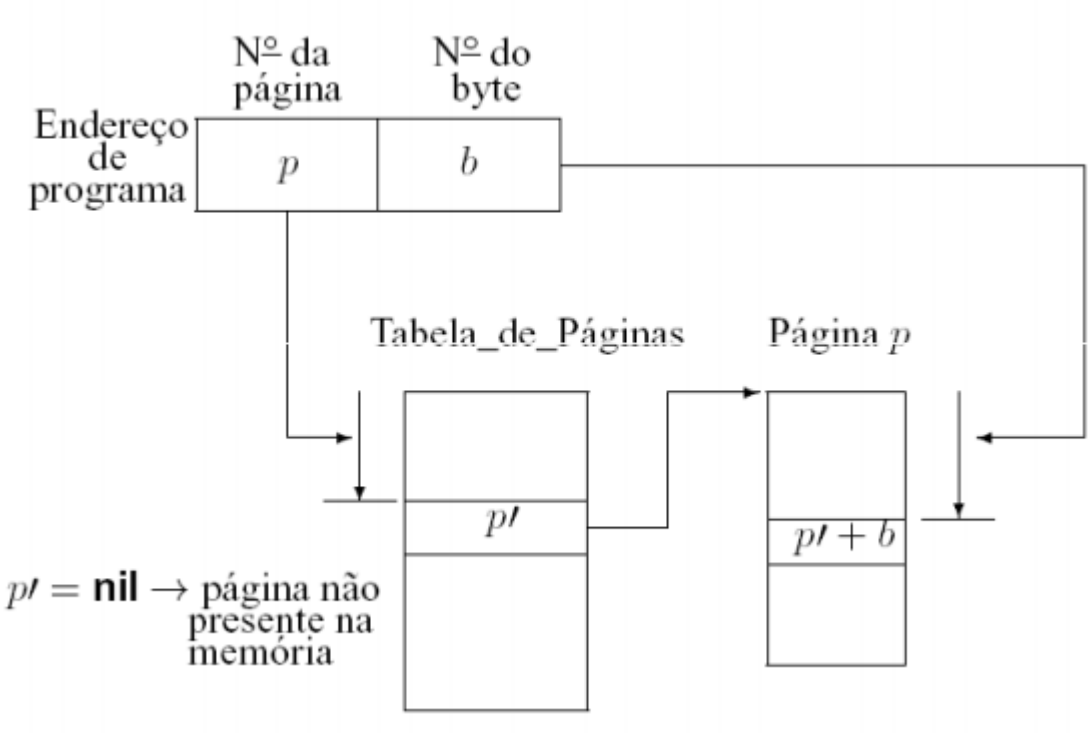
Pesquisa Externa

Pesquisa externa é uma forma de busca em memória secundária (como HDs, e SSDs) , sendo um dos principais motivos para se usar quando o conteúdo desejado não é encontrado em memória principal.

Em comparação com a pesquisa em memória principal, a pesquisa externa é mais lenta, principalmente nos HDs, além de ser um processo bem mais custoso.

Para que o acesso a dados na memória secundária seja facilitado, podemos usar o método de paginação. O endereçamento da memória externa é dividido em páginas de tamanhos iguais, e a memória principal é dividida em molduras de páginas, sendo estas iguais às da memória secundária. Quando um método de pesquisa é usado, a página necessária é carregada para alguma moldura disponível dentro da memória principal, assim podendo ser utilizada.O sistema de paginação tem duas funções principais: Mapeamento de endereços e Transferência de páginas.

Para endereçar um item de uma página, uma parte dos bits do endereço é utilizada para representar o número da página e a outra parte o número do byte do item dentro da página. Já o mapeamento de endereços das memórias principal e secundária é realizado por meio de uma Tabela de Páginas.



Para evitar que a memória principal não tenha páginas vazias, há algumas políticas para substituição das páginas como: LFU, onde a página utilizada com menos frequência será substituída; FIFO, que é como um sistema de filas, onde o primeiro a chegar, é o primeiro a sair e a LRU, onde a página menos utilizada é retirada.

Acesso Sequencial Indexado

O Acesso Sequencial Indexado usa o princípio da pesquisa sequencial, onde cada item é lido sequencialmente até encontrar uma chave maior ou igual a chave de pesquisa, sendo necessário o arquivo estar devidamente ordenado.

O acesso às informações em memória secundária é feito por uma chave que te orienta por uma sequência de valores, onde P é o endereço onde essa chave se encontra e X é o item que contém a chave.

Em um arquivo, cada página tem capacidade para armazenar uma determinada quantidade de itens do disco e cada entrada do índice de páginas armazena a chave do 1º item de cada página e o endereço de tal página no disco.

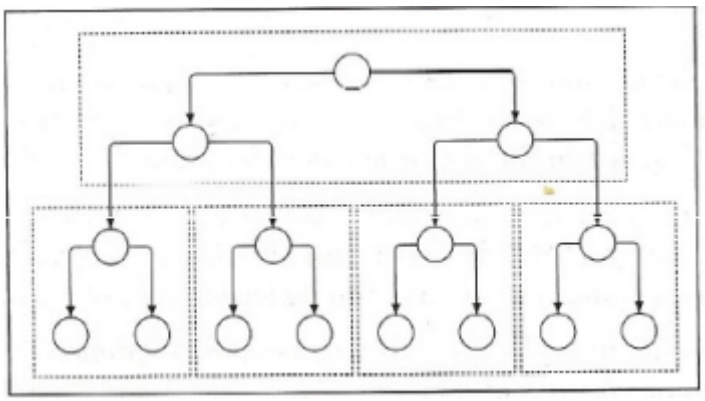
Para se pesquisar por um item, primeiro localiza-se, no índice de páginas, a página que pode conter o item desejado de acordo com sua chave de pesquisa, e depois realiza-se uma pesquisa sequencial na página localizada, até que se encontre ou não a chave desejada.

Árvore Binária de Pesquisa

Árvore Binária de Pesquisa é um método eficiente e rápido de pesquisa, proporcionando um acesso direto e sequencial, proporcionando facilidade na inserção e na retirada de itens, além de boa utilização de espaço de memória.

Para se pesquisar um item em grandes arquivos de dados armazenados em memória secundária, árvores binárias de pesquisa podem ser usadas de forma simplista. Os nodos da árvore são armazenados em disco e os seus apontadores à esquerda e à direita armazenam endereços de disco ao invés de endereços de memória principal.

Para diminuir o número de acessos à memória secundária, os nodos de uma árvore podem ser agrupados em páginas.



A forma na qual os nodos são organizados nas páginas ajuda muito durante a pesquisa, porém uma organização ótima é um problema muito complexo de se resolver.

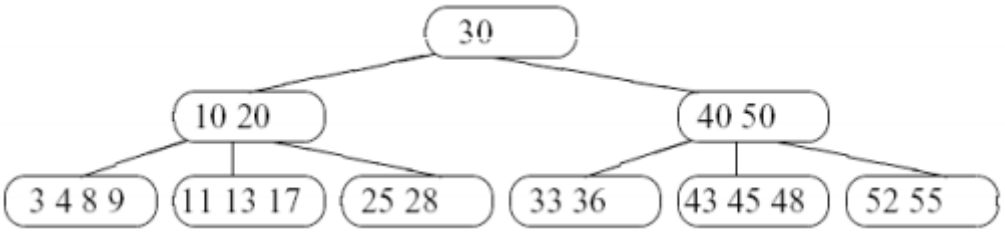
O método de alocação de nodos em páginas, considera a proximidade dos nodos dentro da árvore. O novo nodo é sempre colocado na mesma página do nodo pai. Se a página do nodo pai estiver cheia, o novo nodo é colocado no início de uma nova página criada. Uma das vantagens é que o número de acessos na pesquisa é próximo do ótimo, porém, a ocupação média das páginas é baixa.

Árvore B

A Árvore B veio como uma solução para a árvore binária, onde a árvore fica sempre equilibrada durante o crescimento, e a inserção e a retirada é bem melhor e mais facilitada. O seu funcionamento é bem semelhante à árvore binária, porém cada raiz tem 2 ou mais folhas. As raízes são mais comumente chamadas de páginas.

Em uma Árvore B de ordem m, tem-se a página raiz que contém entre 1 e 2m itens, e as demais páginas, que contém, no mínimo, m itens e m+1 descendentes e, no máximo, 2m itens e 2m+1 descendentes. As páginas folhas aparecem todas no mesmo nível, e os itens que a compõem aparecem dentro de uma página em ordem crescente, de acordo com suas chaves, da esquerda para a direita.

Exemplo de uma Árvore B de ordem m = 2:

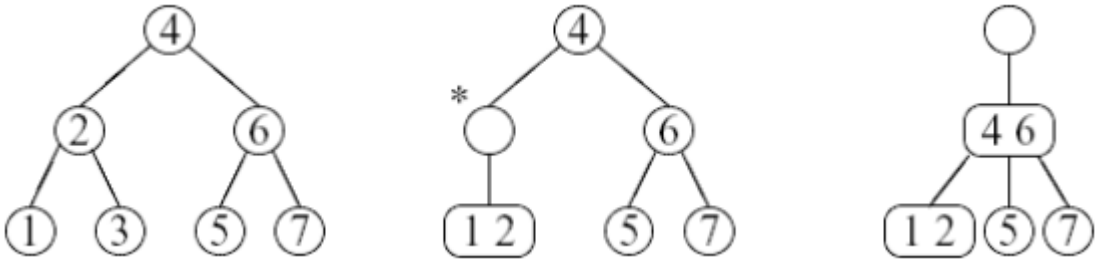


A operação de pesquisa em uma árvore B é semelhante à pesquisa em uma árvore binária de pesquisa. Para encontrar um determinado item, o algoritmo compara a chave do item com as chaves que estão na página raiz até encontrar a chave desejada ou o intervalo no qual ela se encaixa. Caso não tenha localizado a chave desejada, o algoritmo leva o apontador para a subárvore do intervalo encontrado. O processo é repetido recursivamente até que o item procurado seja encontrado ou atingir uma página folha que aponte o final da árvore. O intervalo desejado pode ser encontrado por meio de uma pesquisa sequencial ou algum outro método mais eficiente.

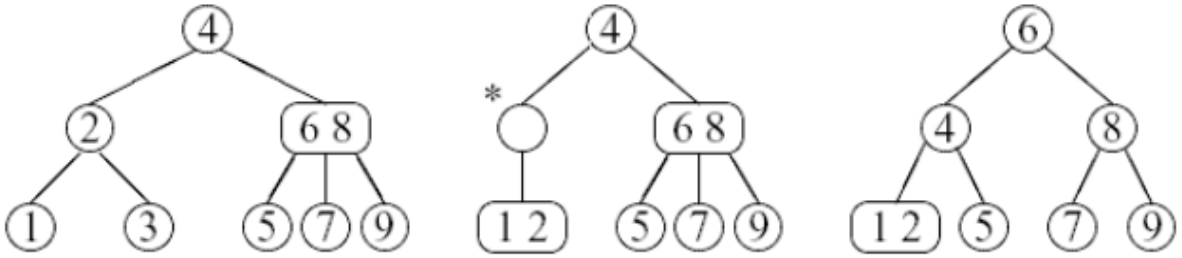
Para inserir um determinado item em uma Árvore B, o algoritmo deve encontrar a página certa, e que respeite as suas propriedades. Se a página a ter o item inserido tiver 2m itens, a inserção será limitada somente a ela, senão teremos um dos processos de balanceamento da árvore, onde uma nova página é criada e os itens são divididos entre essa página nova e a que foi inserido o item e também na página pai. Se a página pai estiver cheia, o processo de divisão se propaga.

A remoção de um item em uma árvore B sempre implica na remoção de um item presente em uma página folha. Caso o item desejado esteja em uma página folha, a remoção é direta. Caso contrário, substitui-se o item desejado pelo item que contenha a chave adjacente antecessora ou sucessora, e então remove-se o item desejado. Assim que um item é removido de uma página folha, é necessário verificar se tal página contém, pelo menos, m itens. Senão, é necessário reconstituir a propriedade da árvore B, tomando emprestado um item da página vizinha. Para isso, temos duas possibilidades: quando a página vizinha possui m itens, e quando tem mais de m itens.

Quando temos m itens na página vizinha, a soma da quantidade de itens tanto na página que vai sofrer a retirada e na vizinha são 2m-1, elas serão fundidas em uma só, pegando da página pai o item do meio, e liberando uma das páginas. Este processo pode propagar até a página raiz e, a tornando vazia, e também a eliminando, causando redução na altura da árvore.

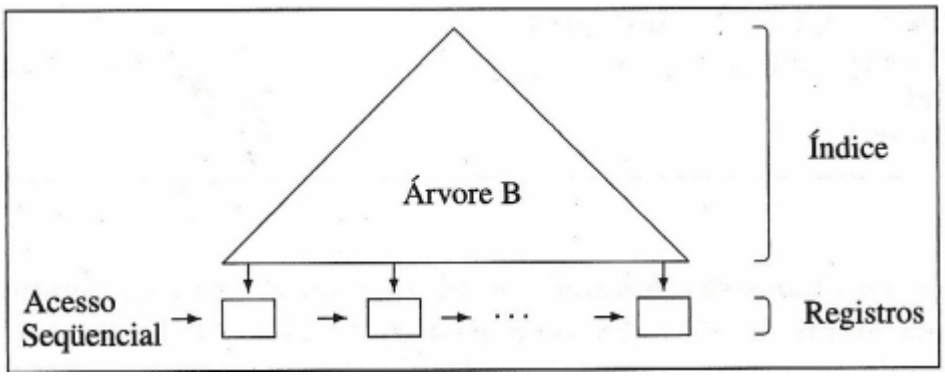


Já quando temos que o número de itens na página vizinha é maior que m, deve-se tomar emprestado um item da página vizinha e trazê-lo para a página em questão via página pai.



Árvore B*

A Árvore B* é uma nova forma de implementação da Árvore B, onde os itens ficam nos níveis mais baixos (em forma de Árvore B) e os níveis mais altos ficam os índices de acesso usando busca sequencial.



Há uma separação lógica entre o índice e os itens que constituem o arquivo, onde no índice contém somente as chaves dos itens, e o resto do arquivo se encontra nas páginas folha. Ademais, as páginas folha podem ou não estar conectadas da esquerda para a direita, permitindo um acesso sequencial mais eficiente do que o acesso via índice.