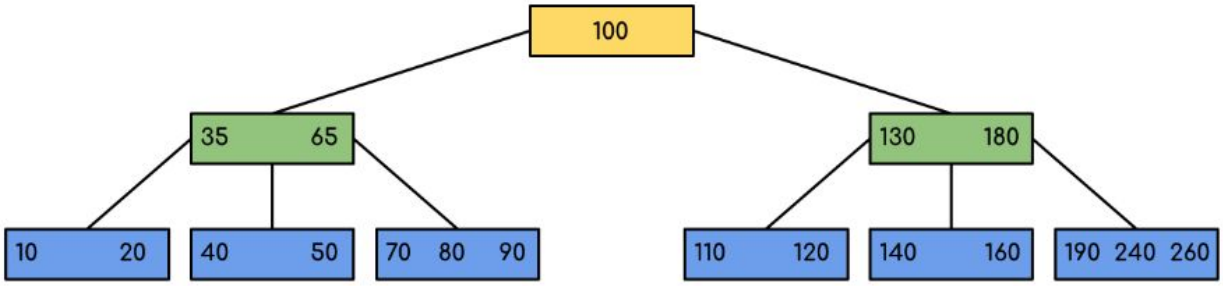
Trabalho 07 - Árvore B - Grupo 12

1 - Quais as propriedades de uma Árvore B?

Para entender o uso de árvores B, devemos pensar na enorme quantidade de dados que não cabe na memória principal, diferente do que acontece com a maioria das outras árvores de pesquisa com auto balanceamento (como árvore AVL e árvore rubro-negra), que presumem-se que tudo esteja na memória principal.

Caso o número de chaves é alto, os dados são lidos do disco na forma de blocos. O tempo de acesso ao disco é muito alto comparado ao tempo de acesso à memória principal. A idéia principal do uso de árvores B é reduzir o número de acessos ao disco. A maioria das operações da árvore (pesquisar, inserir, excluir, max, min, etc) exige acesso O (h) ao disco onde h é a altura da árvore. A altura da árvore B é mantida baixa, colocando o máximo de chaves possíveis em um nó da árvore B. Geralmente, o tamanho do nó é mantido igual ao tamanho do bloco de disco. Como a altura da árvore B é baixa, o acesso total ao disco para a maioria das operações é reduzido significativamente em comparação com as árvores de pesquisa binária equilibradas, como as árvore AVL, rubro-negra, etc.

Abaixo é mostrado um exemplo de árvore B de ordem mínima 5. Observe que, em árvores B práticas, o valor da ordem mínima é muito maior que 5.



É possível notar que as folhas estão no mesmo nível, e todas as não folhas não possuem uma sub-árvore vazia e possuem chaves uma a menos que o número de seus filhos.

As propriedades da Árvore B são:

-Todas as chaves de um nó são classificadas em ordem crescente. O filho entre duas chaves k1 e k2 contém todas as chaves no intervalo de k1 e k2.

-Uma árvore B é definida pelo termo grau mínimo 't'. O valor de t depende do tamanho do bloco de disco.

-Cada nó, exceto a raiz, deve conter pelo menos t-1 chaves. A raiz pode conter no mínimo 1 chave.

-Todas as folhas estão no mesmo nível.

-Todos os nós (incluindo raiz) podem conter no máximo 2t - 1 chaves.

-O número de filhos de um nó é igual ao número de chaves nele mais 1.

-A Árvore B cresce e encolhe a partir da raiz, diferente da Árvore de Pesquisa Binária. As árvores de pesquisa binária crescem para baixo e também encolhem para baixo.

-Como outras árvores de pesquisa binária equilibradas, a complexidade do tempo para pesquisar, inserir e excluir é O (log n).

2 - Como garantir essas propriedades?

Ao executar algumas operações na Árvore B, qualquer propriedade pode ser violada, como o número mínimo de filhos que um nó pode ter. Para manter as propriedades da Árvore B, a árvore pode ser dividida ou unida.

Nas árvores B, os nós internos (sem folha) podem ter um número variável de nós filhos dentro de um intervalo predefinido. Quando dados são inseridos ou removidos de um nó, seu número de nós filhos é alterado. Para manter o intervalo predefinido, os nós internos podem ser unidos ou divididos. Como uma variedade de nós filhos é permitida, as árvores B não precisam ser reequilibradas com tanta freqüência quanto as outras árvores de pesquisa com auto balanceamento, mas podem desperdiçar algum espaço, pois os nós não estão totalmente cheios. Os limites inferior e superior do número de nós filhos geralmente são corrigidos para uma implementação específica. Por exemplo, em uma árvore B de 2-3 (geralmente chamada de árvore 2-3), cada nó interno pode ter apenas 2 ou 3 nós filhos.

As Inserções e exclusões em uma árvore B devem manter essas propriedades. Ao fazer isso, o caminho mais longo em uma árvore B com n chaves contém no máximo nós log m (n), levando ao desempenho O(log n) de suas operações. Árvores B podem ser prontamente armazenadas em memórias secundárias aumentando o número de chaves em cada nó para que seu tamanho total seja adequadamente alinhado com o tamanho da página (por exemplo, armazenando dois nós da Árvore B por página), o que minimiza o número de leituras de disco para carregar os nós na memória primária.

3 - Em que contexto as Árvores B são mais úteis?

É mais útil quando se está interagindo com algum componente externo de memória e o tempo para acessar está informação no nó supera o tempo de processamento desta informação. Imagine um cenário onde existem milhões de pontos de informações e você não pode carregar todos eles na memória e toda vez que você precisar acessar o filho daquele nó, você precisa lê-lo através do disco rígido, e é importante lembrar que a leitura em disco leva um tempo muito alto, para essas ocasiões todo essas operações complicadas de uma Árvore B levam um tempo inferior e proporcionam um melhor rendimento na busca e coleta de dados, pois a árvore binária possui um tempo de busca de O(log(n)).

Este tipo interação acontece como por exemplo em banco de dados e estruturas de arquivos, onde possuem diversos valores e chaves para acesso.

4 - Quais as principais variantes das Árvores B, e suas características?

Principais variantes da árvore B

As principais variantes da árvore B são definidas como, árvore B* e árvore B+. As árvores B são uma estrutura de dados que se aplicam a uma manipulação de um grande volume de dados, proporciona rápido acesso aos dados e possui custo mínimo de overhead. Existe variantes desta que por vez, correspondem certas características como as árvores B* e B+.

Árvore B+

A árvore B+ é uma uma estrutura de dados, uma variação da estrutura básica da árvore B, porém ela tem um armazenamento diferente de suas chaves no uso de algoritmos de inserção, remoção e busca. Na árvore B+, os registros podem ser armazenados apenas nos nós das folhas, enquanto os nós internos podem armazenar apenas os valores da chave, enquanto na árvore B as chaves e os registros podem ser armazenados nos nós internos e folhas.

Características:

- Todas as chaves estão armazenadas nas folhas, há uma ligação entre os irmãos folhas adjacentes para facilitar a busca.
- As folhas são encadeadas.
- Permite uma facilidade na leitura sequencial, as folhas são ligadas, cada folha aponta para a próxima folha.
- Não é necessário movimentar os nós quando tiver que fazer a manutenção da ordenação de dados.
- As chaves se repetem em nós não folhas que acabam formando um índice.
- Somente as chaves são mantidas nos nós não terminais, no entanto obtém-se árvores com maior número de entradas por nó e menos profundidade.
- Os links que mantém os nós folhas dispostos em sequência, garante o processamento linear.

Árvore B*

A árvore B* é uma estrutura de dados, uma variação da estrutura básica da árvore B, apesar das semelhanças, durante as operações de inserção a técnica de redistribuição de chaves também é empregado. Para que as folhas adjacentes estejam completamente cheias, ela posterga a divisão de nós. E somente quando as folhas estiverem cheias ela realiza o particionamento de duas páginas irmãs, desde que haja a redistribuição de chaves entre as páginas, seria como uma estratégia dessa variação.

Características:

- Cada nó contém no mínimo, ⅔ do número máximo de chaves
- Até que as duas páginas irmãs estejam cheias, a subdivisão é adiada.
- Na sequência, a divisão do conteúdo das duas páginas em três páginas, é realizada.

Referências:

https://www.javatpoint.com/b-tree
http://www.r-5.org/files/books/computers/algo-list/common/Heineman_Pollice_Selkow-Algorithms_in_a_Nutshell-EN.pdf
https://www.youtube.com/watch?v=aZjYr87r1b8
https://www.cpp.edu/~ftang/courses/CS241/notes/b-tree.htm#:~:text=A%20B%2Dtree%20is%20a,in%20database%20and%20file%20systems.
https://www.youtube.com/watch?v=C_q5ccN84C8
http://wiki.icmc.usp.br/images/8/8e/SCC578920131-B.pdf
https://marciobueno.com/arquivos/ensino/ed2/ED2_04_Arvore_B+.pdf