



Algoritmo e Estrutura de Dados II

COM-112

B-Tree

Vanessa Souza



Árvores

▶ Motivação

- ▶ Quando não conseguimos trabalhar na memória principal (ou primária), temos que usar a memória secundária...
- ▶ Sabemos que o acesso aos dados em memória secundária é muito lento.
- ▶ Precisamos de meios eficientes de acesso aos dados (provavelmente na forma de “índices”)

FONTE : Prof. Dr. Ednaldo Pizzolato





Árvores

▶ Motivação :

- ▶ Assuma que um disco gire a 3600 RPM
- ▶ Em 1 minuto faz 3.600 rotações, portanto uma rotação leva $1/60$ de segundo, ou 16.7ms
- ▶ Na média cada acesso gastaria 8ms
- ▶ Parece ok até nos darmos conta que 120 acessos a disco consomem um segundo – o mesmo que 25 milhões de instruções
- ▶ Ou seja, um acesso a disco é equivalente a 200.000 instruções

FONTE : Prof. Dr. Ednaldo Pizzolato





Razões para usar árvores B

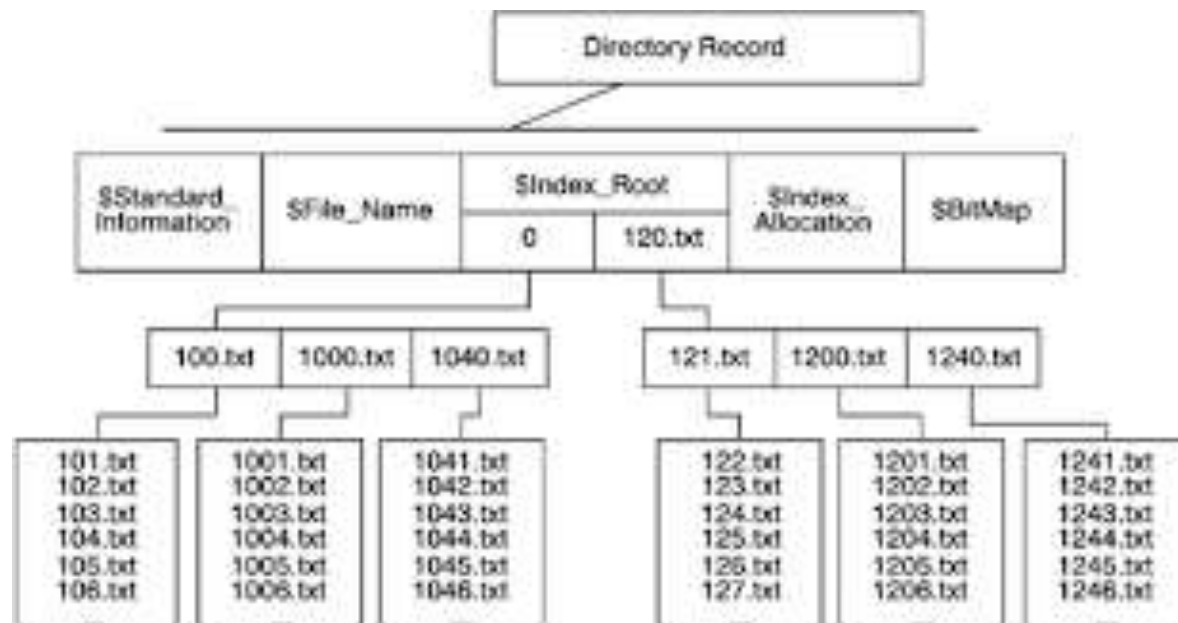
- ▶ Árvores B são a estrutura subjacente a muitos sistemas de arquivos e bancos de dados. Por exemplo:
 - ▶ o sistema de arquivos NTFS do Windows,
 - ▶ o sistema de arquivos HFS do Mac,
 - ▶ os sistemas de arquivos ReiserFS, XFS, Ext3FS, JFS do Linux,
 - ▶ e os bancos de dados ORACLE, DB2, INGRES, SQL e PostgreSQL.





Razões para usar árvores B

- ▶ Árvores B são a estrutura subjacente a muitos sistemas de arquivos e bancos de dados. Por exemplo:
 - ▶ o sistema de arquivos NTFS do Windows
 - ▶ Um diretório é implementado como uma B-Tree de nomes de arquivos





Ordem da B-TREE

- ▶ Vimos que a ordem (m) de uma b-tree nesse curso será definida como a quantidade **MÁXIMA DE FILHOS** que um nó pode ter.
 - ▶ Se $m = 5$
 - ▶ O nó pode ter, no máximo, 5 filhos
 - ▶ O nó terá, no máximo, 4 chaves
 - ▶ O nó terá, no mínimo, 2 chaves e, portanto, 3 filhos
 - Propriedade da B-TREE
 - Ocupação dos nós deve ser, pelo menos, metade de sua capacidade





Ordem da B-TREE

- ▶ Vimos que a ordem (m) de uma b-tree nesse curso será definida como a quantidade **MÁXIMA DE FILHOS** que um nó pode ter.
 - ▶ Dado m
 - ▶ O nó terá, no máximo, **$m-1$** chaves
 - ▶ O nó terá, no mínimo, $\left\lceil \frac{m}{2} \right\rceil - 1$ chaves
 - ▶ O nó pode ter, no máximo, **m** filhos
 - ▶ O nó terá, no mínimo, $\left\lceil \frac{m}{2} \right\rceil$ filhos
 - ▶ O meio = $\left\lfloor \frac{m-1}{2} \right\rfloor$





Ordem da B-TREE

- ▶ Vimos também que existem duas variantes do algoritmo de inserção:
 - ▶ Inserção tradicional (*preemptive split*)
 - ▶ EXEMPLO:
 - ▶ Dada uma b-tree de ordem 5, inserir os seguintes valores na ordem: 15, 5, 20, 10, 2



Ordem da B-TREE

- ▶ Vimos também que existem duas variantes do algoritmo de inserção:
 - ▶ Inserção tradicional (*preemptive split*)
 - ▶ EXEMPLO:
 - ▶ Dada uma b-tree de ordem 5, inserir os seguintes valores na ordem: 15, 5, 20, 10, 2
- ▶ Sendo assim, definimos que para a inserção tradicional, **m é sempre PAR**
 - ▶ Pois assim, a quantidade de chaves é ímpar e o balanceamento fica garantido





Ordem da B-TREE

- ▶ Vimos também que existem duas variantes do algoritmo de inserção:
 - ▶ Inserção tradicional (*preemptive split*)
 - ▶ EXEMPLO:
 - ▶ Dada uma b-tree de ordem 5, inserir os seguintes valores na ordem: 15, 5, 20, 10, 2
- ▶ Sendo assim, definimos que para a inserção tradicional, **m é sempre PAR**
 - ▶ Pois assim, a quantidade de chaves é ímpar e o balanceamento fica garantido
 - ▶ <https://www.cs.usfca.edu/~galles/visualization/BTree.html>





Árvore B de Ordem m

- ▶ t é chamado grau mínimo da árvore
 - ▶ Número máximo de CHAVES no nó : $2t-1$
 - ▶ Número mínimo de CHAVES no nó : $t-1$





Exercício

- ▶ Mostre os resultados de inserir as chaves a seguir em uma árvore B de ordem 5 inicialmente vazia:
- ▶ F, S, Q, K, C, L, H, T, V, W, M, R, N, P, A, B, X, Y, D, Z, E