

Árvore B

Profa. Dra. Cristina Dutra de Aguiar Ciferri

Desempenho da Árvore-B

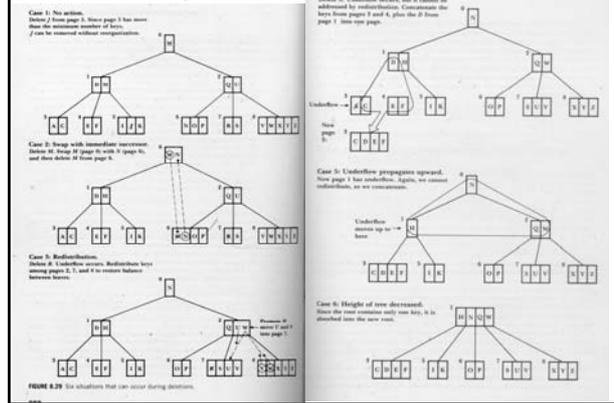
- Baseado em suas propriedades
 - 2 cada página, exceto a raiz e as folhas, possui no mínimo $\lfloor m/2 \rfloor$ descendentes → taxa de ocupação
 - 5 uma página interna com k descendentes contém k-1 chaves
 - 6 uma folha possui no mínimo $\lceil m/2 \rceil - 1$ chaves e no máximo m - 1 chaves → taxa de ocupação

habilidade de garantir que a árvore seja "larga e rasa" ao invés de "estreita e profunda"

Desempenho da Árvore-B

- **Split**
 - garante a manutenção das propriedades da árvore-B durante a inserção de novas chaves
- **Remoção**
 - também deve garantir as propriedades durante a remoção de chaves
 - *underflow*
 - ocorre quando o número de chaves em uma página fica abaixo do número mínimo de chaves permitido pela árvore-B

Remoção: Diferentes Casos



Remoção: Caso 1

- Remoção de uma chave em um nó folha, sem causar *underflow*
 - situação mais simples possível
- Solução
 - eliminar a chave da página
 - rearranjar as chaves remanescentes dentro da página para fechar o espaço liberado

Remoção: Caso 2

- Remoção de uma chave em um nó não folha
- Solução
 - sempre remover chaves somente nas folhas
- Passos
 - trocar a chave a ser removida com a sua chave sucessora imediata (que está em um nó folha)
 - remover a chave diretamente do nó folha

Remoção: Caso 3

- Remoção de uma chave em um nó, causando *underflow*
- Solução: Redistribuição
 - procurar uma página irmã (i.e., que possui o mesmo pai) adjacente que contenha mais chaves do que o mínimo
 - se encontrou
 - redistribuir as chaves entre as páginas
 - reacodar a chave separadora, modificando o conteúdo do nó pai

Algoritmos e Estruturas de Dados II

Árvore-B

Remoção: Caso 4

- Remoção de uma chave em um nó, causando *underflow* e a redistribuição não pode ser aplicada
- Solução: Concatenação
 - combinar para formar uma nova página
 - o conteúdo do nó que sofreu *underflow*
 - o conteúdo de um nó irmão adjacente
 - a chave separadora no nó pai
 - tratar o *underflow* no nó pai, caso necessário

Algoritmos e Estruturas de Dados II

Árvore-B

Concatenação

- Processo inverso do *split*
 - Características
 - reverte a promoção de uma chave
 - pode causar *underflow* no nó pai
- ⇒ concatenação pode ser propagada em direção ao nó raiz

ocorre a redução no número total de nós da árvore

Algoritmos e Estruturas de Dados II

Árvore-B

Remoção: Caso 5

- *Underflow* no nó pai causado pela remoção de uma chave em um nó filho
- Solução
 - utilizar redistribuição ou concatenação, dependendo da quantidade de chaves que a página irmã adjacente contém

Algoritmos e Estruturas de Dados II

Árvore-B

Remoção: Caso 6

- Diminuição da altura da árvore
- Característica
 - o nó raiz possui uma única chave
 - a chave é absorvida pela concatenação de seus nós filhos
- Solução
 - eliminar a raiz antiga
 - tornar no nó resultante da concatenação dos nós filhos a nova raiz da árvore

Algoritmos e Estruturas de Dados II

Árvore-B

Remoção em Árvore-B

1. se a chave a ser removida não estiver em um nó folha, troque-a com sua sucessora imediata, que está em um nó folha
2. remova a chave
3. após a remoção, se o nó satisfaz o número mínimo de chaves, nenhuma ação adicional é requerida

Algoritmos e Estruturas de Dados II

Árvore-B

Remoção em Árvore-B

4. após a remoção, caso ocorra *underflow*, verifique o número de chaves nos nós irmãos adjacentes à esquerda e à direita
 - a. se algum nó irmão adjacente possui mais do que o número mínimo de chaves, aplique a redistribuição
 - b. se nenhum nó irmão adjacente possui mais do que o número mínimo de chaves, aplique a concatenação

Algoritmos e Estruturas de Dados II

Árvore-B

Remoção em Árvore-B

5. se ocorreu concatenação, repita os passos 3 a 5 para o nó pai
6. se a última chave da raiz for removida, a altura da árvore é diminuída

Algoritmos e Estruturas de Dados II

Árvore-B

Redistribuição

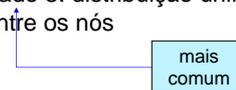
- Representa uma idéia inovadora
 - diferente do *split* ou da concatenação
- Não se propaga para os nós superiores
 - apenas efeito local na árvore
- Baseada no conceito de nós irmãos adjacentes
 - dois nós logicamente adjacentes, mas com pais diferentes não são irmãos

Algoritmos e Estruturas de Dados II

Árvore-B

Redistribuição

- Não fixa a forma na qual as chaves devem ser redistribuídas
 - possibilidade 1: mover somente uma chave, mesmo que a distribuição das chaves entre as páginas não seja uniforme
 - possibilidade 2: mover k chaves
 - possibilidade 3: distribuição uniforme das chaves entre os nós



Algoritmos e Estruturas de Dados II

Árvore-B

Redistribuição durante Inserção

- Funcionalidade
 - permite melhorar a taxa de utilização do espaço alocado para a árvore
- *split*
 - divide uma página com *overflow* (i.e., *working page*) em duas páginas semi-vazias (i.e., *page* e *newpage*)
- redistribuição
 - a chave que causou *overflow* (além de outras chaves) pode ser colocada em outra página

Algoritmos e Estruturas de Dados II

Árvore-B

Redistribuição durante Inserção

- Opção interessante
 - a rotina de redistribuição já está codificada para prover suporte à remoção
 - a redistribuição evita, ou pelo menos adia, a criação de novas páginas
 - tende a tornar a árvore-B mais eficiente em termos de utilização do espaço em disco
 - garante um melhor desempenho na busca, desde que um número menor de nós pode reduzir a altura da árvore, por exemplo

Algoritmos e Estruturas de Dados II

Árvore-B

Split x Redistribuição

- Somente *split* na inserção
 - no pior caso, a utilização do espaço é de cerca de 50%
 - em média, para árvores grandes, o índice de ocupação é de ~69%
- Com redistribuição na inserção
 - em média, para árvores grandes, o índice de ocupação é de ~86%