



Algoritmo e Estrutura de Dados II

COM-112

Aula 17

Vanessa Souza



Árvore B de Ordem m

- ▶ Cada página tem no máximo **m** descendentes
- ▶ A **raiz** tem pelo menos **2** descendentes
- ▶ Todas as folhas estão no mesmo nível
- ▶ O número **máximo** de elementos em um nó é $m-1$
- ▶ O número **mínimo** de elementos em um nó é $\left\lfloor \frac{m}{2} \right\rfloor - 1$
- ▶ A ordem de uma árvore B é escolhida de forma que o tamanho de um nó cheio seja menor que o tamanho de bloco de disco, mas o mais próximo que for possível
- ▶ Existem diversas variantes de árvore B





Árvores B

- ▶ Árvore B+ e B* são **variações** da árvore B



Árvores B*



Árvores B*

- ▶ Proposta por Knuth em 1973
- ▶ Uma árvore B* possui as mesmas propriedades de uma árvore B, mais a seguinte propriedade:
 - ▶ Exige-se que todos os nós, exceto a raiz, estejam pelo menos **2/3 cheios** (em vez de 1/2 cheios).
- ▶ Posterga o *split*
 - ▶ estende a noção de **redistribuição** durante a inserção para incluir novas regras para o particionamento de nós.





Redistribuição

- ▶ Conceito de 'rotação' da remoção em árvores B.
- ▶ Representa uma ideia 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





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**



Redistribuição

- ▶ A redistribuição durante a inserção permite melhorar a taxa de utilização do espaço alocado para a árvore
 - ▶ 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 já que pode reduzir a altura da árvore, por exemplo
- ▶ Split
 - ▶ divide uma página com *overflow* em duas páginas semi-vazias.
- ▶ Redistribuição
 - ▶ a chave que causou *overflow* (além de outras chaves) pode ser colocada em outra página





Árvore B*

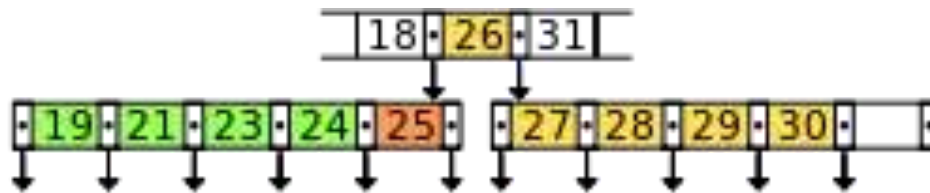
- ▶ Cada página da árvore deve conter no mínimo 2/3 de chaves
 - ▶ Dada uma árvore B* de ordem m
 - ▶ Nro_máximo de chaves no nó: $m-1$
 - ▶ Nro_mínimo de chaves no nó : $\left\lfloor \frac{2m-1}{3} \right\rfloor$
- ▶ ***Para conseguir isto, o algoritmo deve executar sempre a redistribuição de chaves entre duas páginas irmãs até ambas ficarem cheias.***
- ▶ Somente neste caso haverá uma divisão de páginas.
- ▶ Mas, ao invés de duas, três páginas com 2/3 chaves serão geradas
 - ▶ Split 2-to-3



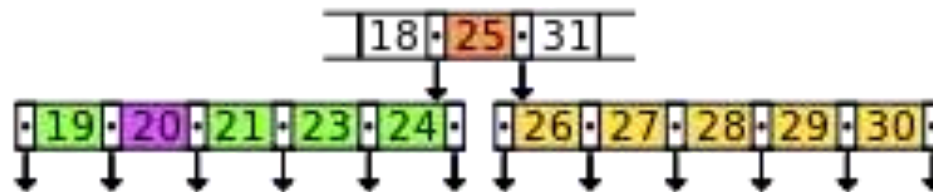


Árvore B*

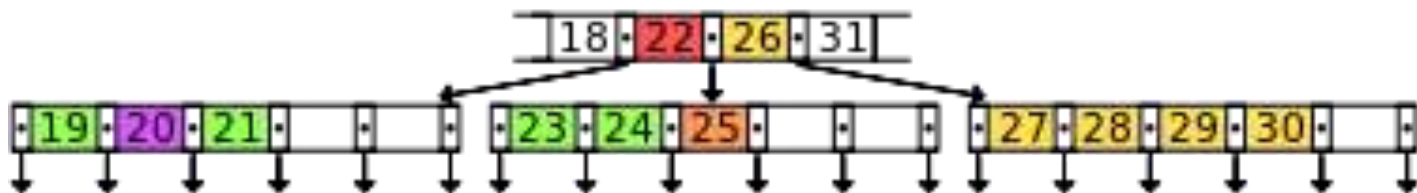
m=6



INSERT 20



INSERT 22

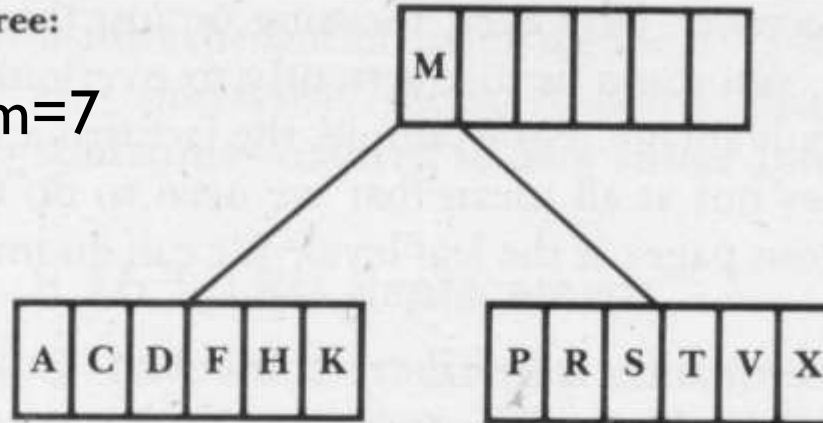




Árvore B*

Original tree:

$m=7$



Two-to-three-split:

After the insertion of the key *B*.

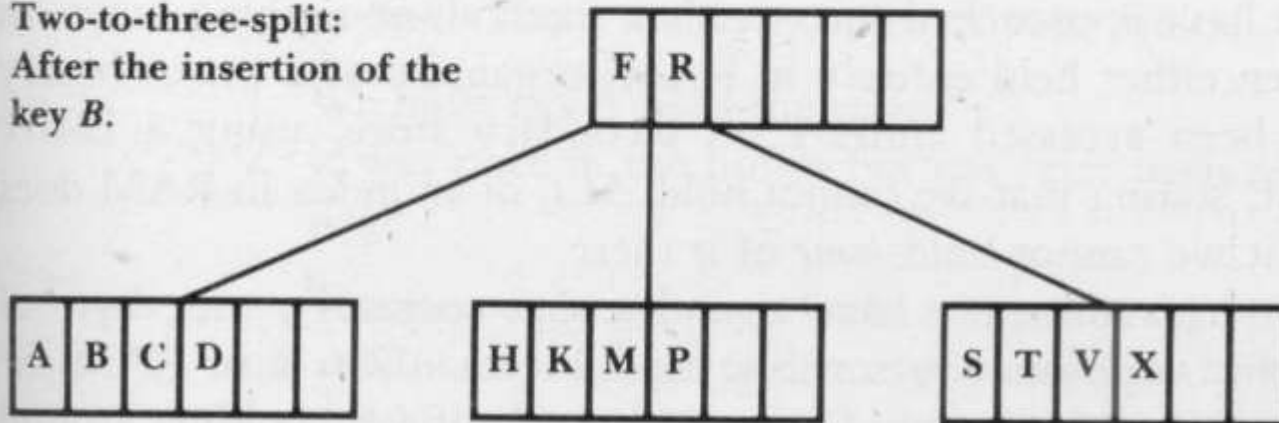


FIGURE 8.30 A two-to-three split.



Árvore B*

- ▶ 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 $\approx 69\%$
- ▶ Com redistribuição na inserção
 - ▶ em média, para árvores grandes, o índice de ocupação é de $\approx 86\%$





Árvore B*

- ▶ Mudança na taxa de ocupação
 - ▶ afeta as rotinas de remoção e redistribuição
- ▶ Particionamento da raiz
 - ▶ Problema
 - ▶ raiz não possui nó irmão
 - ▶ Soluções
 - ▶ dividir a raiz usando a divisão convencional (1-to-2 split);
ou
 - ▶ permitir que a raiz seja maior





Árvores B+



Árvores B

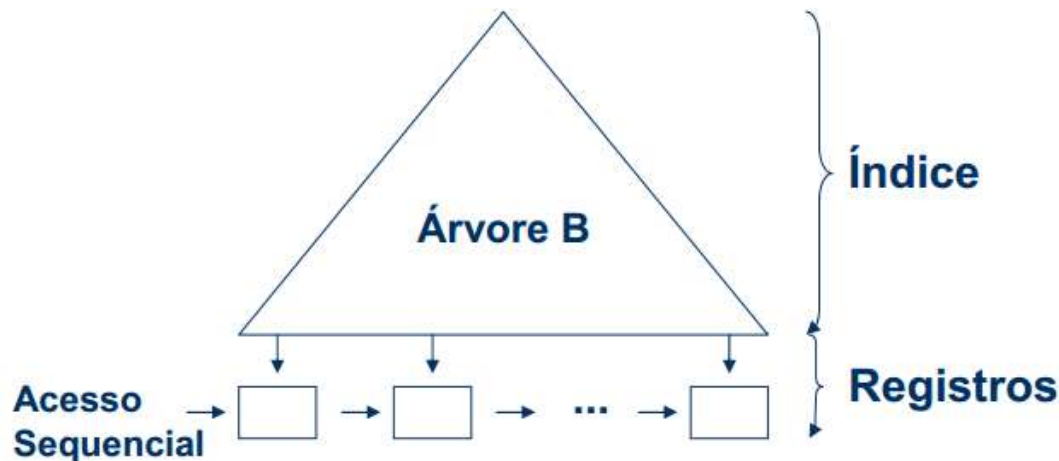
- ▶ Uma árvore B+ possui algumas características diferentes da árvore B:
 - ▶ Em uma árvore B+, todos os registros são armazenados no último nível (páginas folhas)
 - ▶ Os níveis acima do último nível constituem um índice cuja organização é a organização de uma árvore B



Árvores B

- ▶ Uma árvore B+ possui algumas características diferentes da árvore B:

Separação lógica entre o índice e os registros que constituem o arquivo



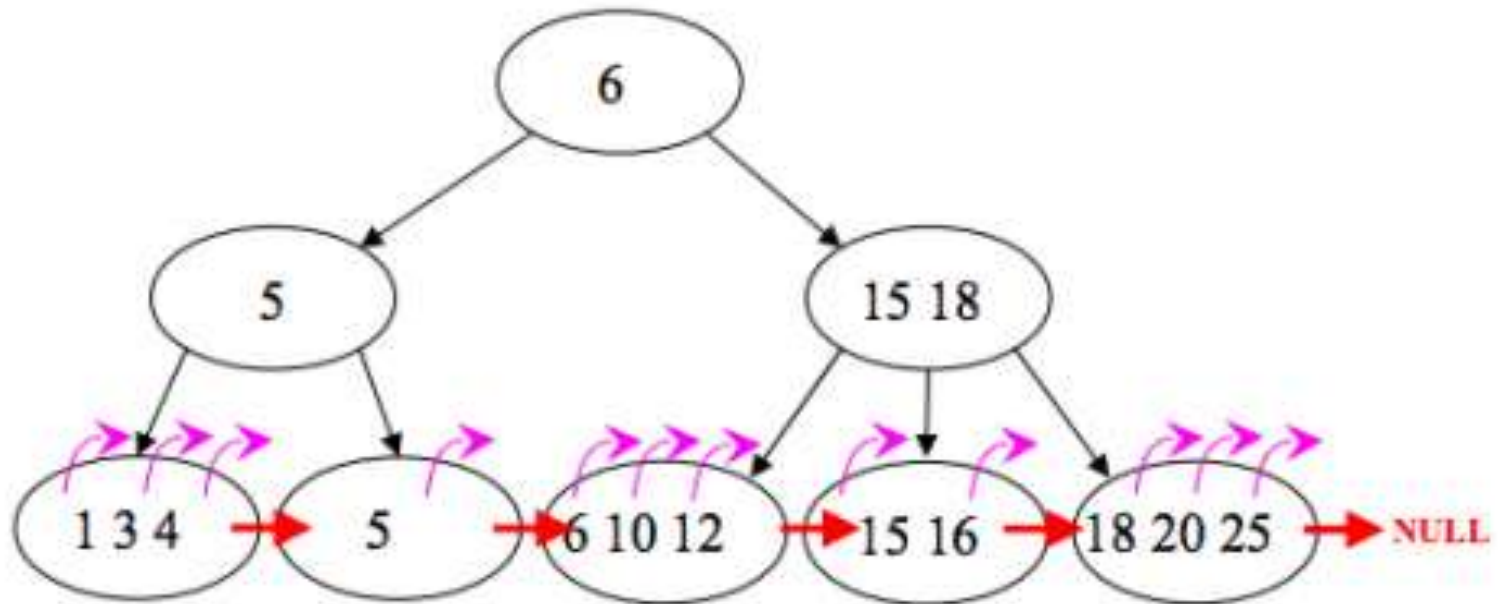


Árvores B+

- ▶ A principal diferença é que na árvore B+, os nós intermediários (índice) não possuem ponteiros para dados, mas apenas as chaves.
- ▶ As folhas são interligadas



Árvores B+





Exemplo - Relação

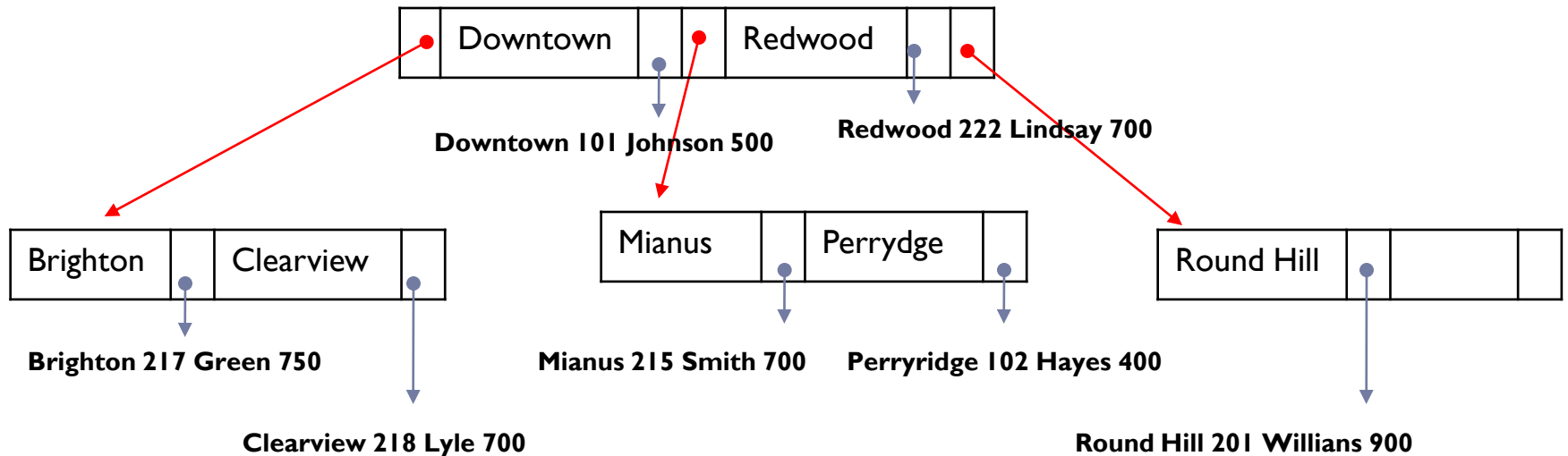
- Relação depósito no banco de dados bancário.

| Registro | Nome-agência | Numero-conta | Nome-cliente | saldo |
|-----------------|---------------------|---------------------|---------------------|--------------|
| 0 | Perryridge | 102 | Hayes | 400 |
| 1 | Round Hill | 305 | Turner | 350 |
| 2 | Mianus | 215 | Smith | 700 |
| 3 | Downtown | 101 | Johnson | 500 |
| 4 | Redwood | 222 | Lindsay | 700 |
| 5 | Round Hill | 201 | Williams | 900 |
| 6 | Brighton | 217 | Green | 750 |
| 7 | Clearview | 218 | Lyle | 700 |





Exemplo – Árvore B



A árvore B gerencia o espaço usado por seus blocos para que eles sempre estejam ocupados com pelo menos a metade de sua capacidade.



Exemplo - Relação

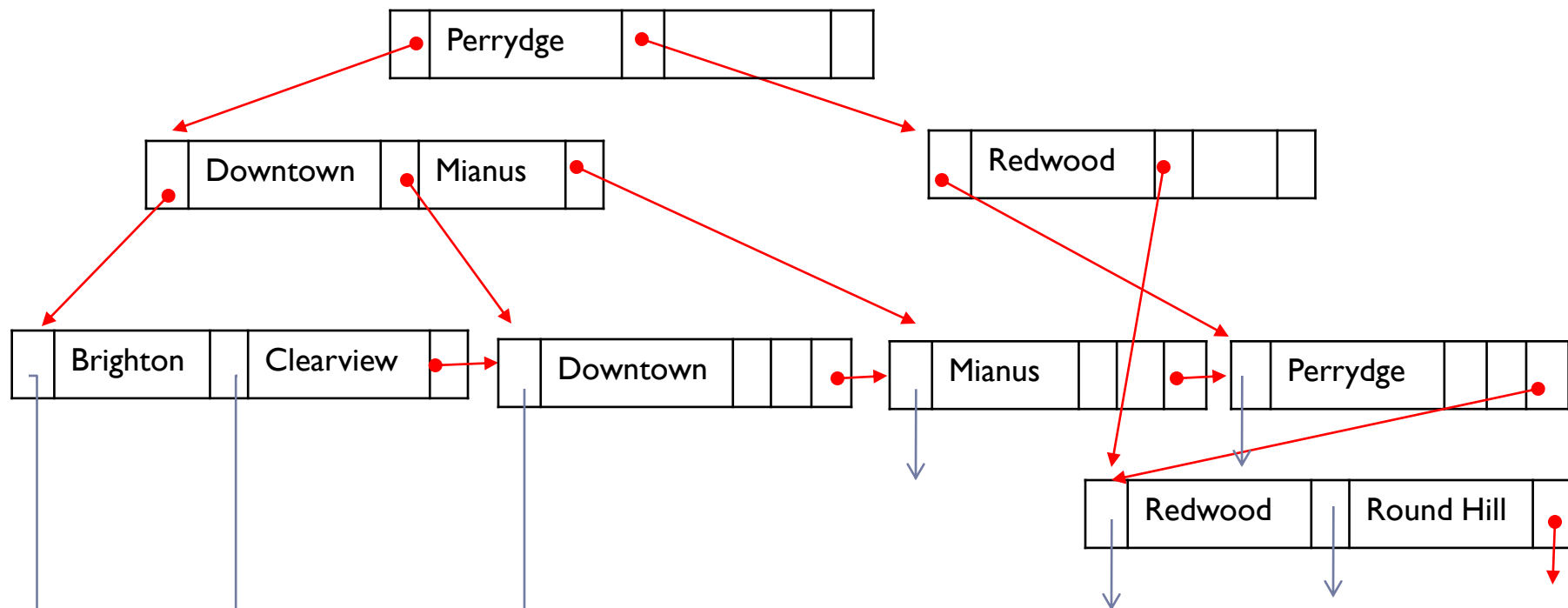
- Relação depósito no banco de dados bancário.

| Registro | Nome-agência | Numero-conta | Nome-cliente | saldo |
|-----------------|---------------------|---------------------|---------------------|--------------|
| 0 | Perryridge | 102 | Hayes | 400 |
| 1 | Round Hill | 305 | Turner | 350 |
| 2 | Mianus | 215 | Smith | 700 |
| 3 | Downtown | 101 | Johnson | 500 |
| 4 | Redwood | 222 | Lindsay | 700 |
| 5 | Round Hill | 201 | Williams | 900 |
| 6 | Brighton | 217 | Green | 750 |
| 7 | Clearview | 218 | Lyle | 700 |





Exemplo – Árvore B+



| Nome-agência | Numero-conta | Nome-cliente | saldo |
|--------------|--------------|--------------|-------|
| Brighton | 217 | Green | 750 |
| Clearview | 218 | Lyle | 700 |
| Downtown | 101 | Johnson | 500 |
| Mianus | 215 | Smith | 700 |
| Perryridge | 102 | Hayes | 400 |
| Redwood | 222 | Lindsay | 700 |
| Round Hill | 305 | Turner | 350 |



Árvore B+

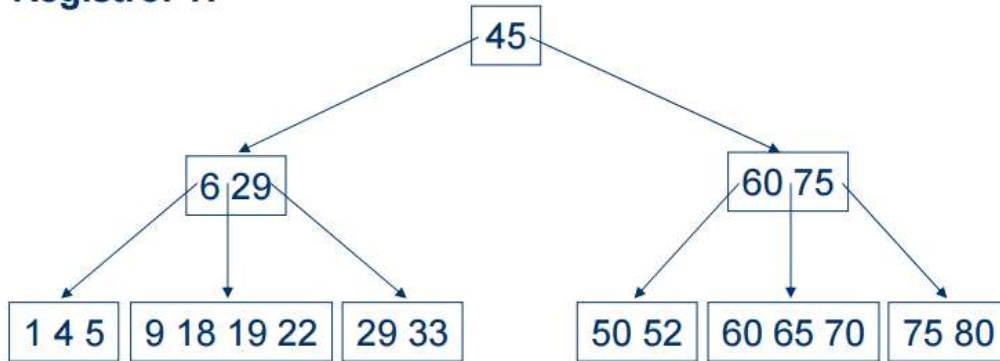
- ▶ A separação lógica afeta as rotinas de inserção e remoção
- ▶ Inserção
 - ▶ Quando uma folha é dividida em duas, o algoritmo promove uma cópia da chave que pertence ao registro do meio para a página pai no nível anterior, retraindo o registro do meio na página folha da direita



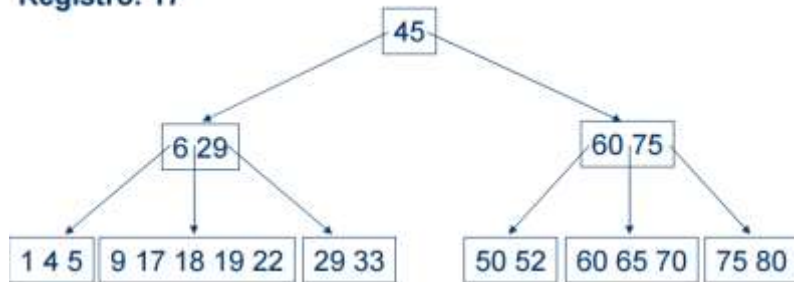


Árvore B+ -- Inserção

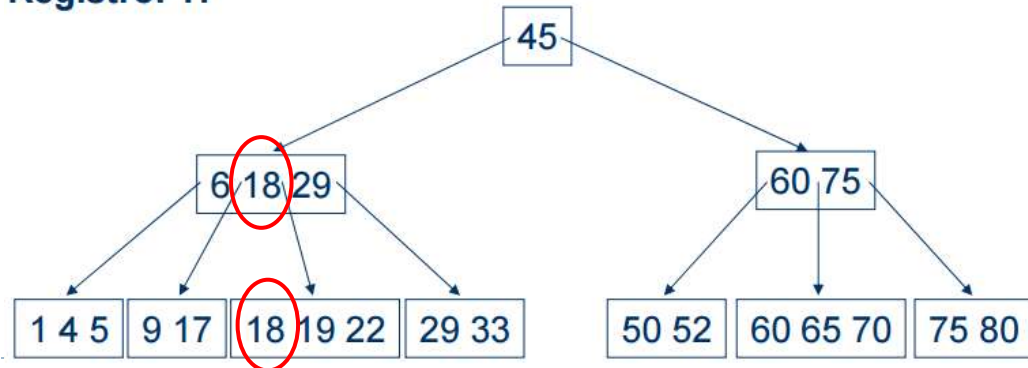
Registro: 17



Registro: 17



Registro: 17





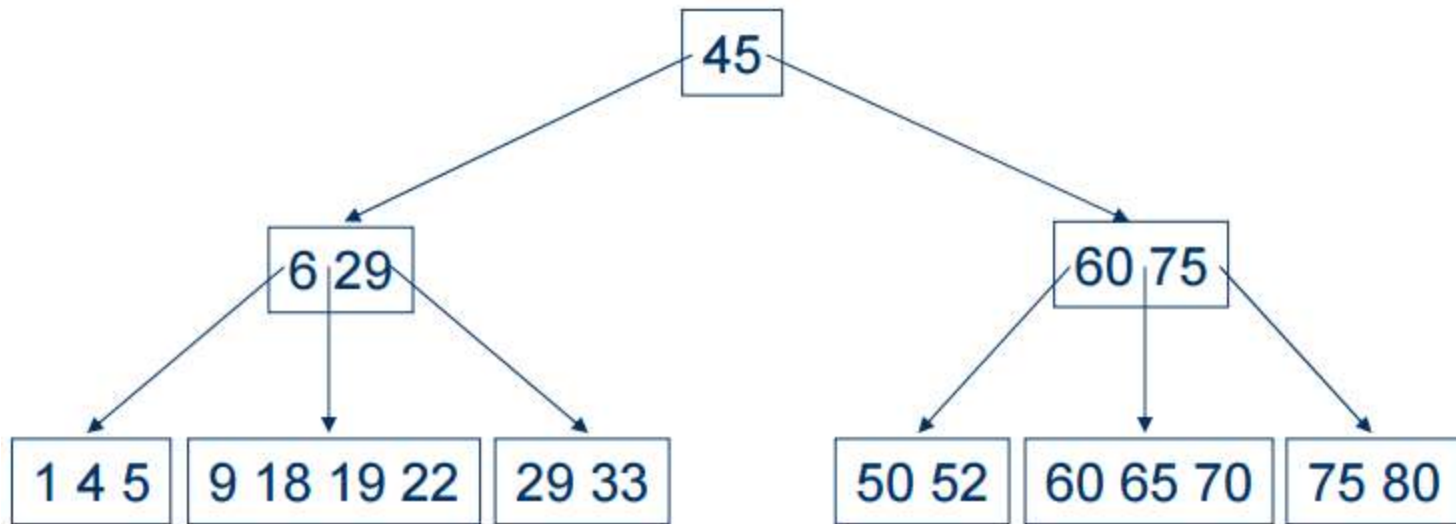
Árvore B+

- ▶ A separação lógica afeta as rotinas de inserção e remoção
- ▶ Remoção
 - ▶ Mais simples do que a remoção de uma árvore B.
 - ▶ O registro a ser removido reside sempre em uma página folha, o que torna sua remoção simples
 - ▶ Tratar apenas a cópia do pai para a folha



Árvore B+

Remover: 5, 19, 22, 60 e 9





Vantagens da Árvore B sobre a B+

- ▶ Ausência de armazenamento redundante de chaves de busca;
- ▶ Possibilidade de encontrar uma chave sem chegar até um nó folha;
 - ▶ Busca mais rápida

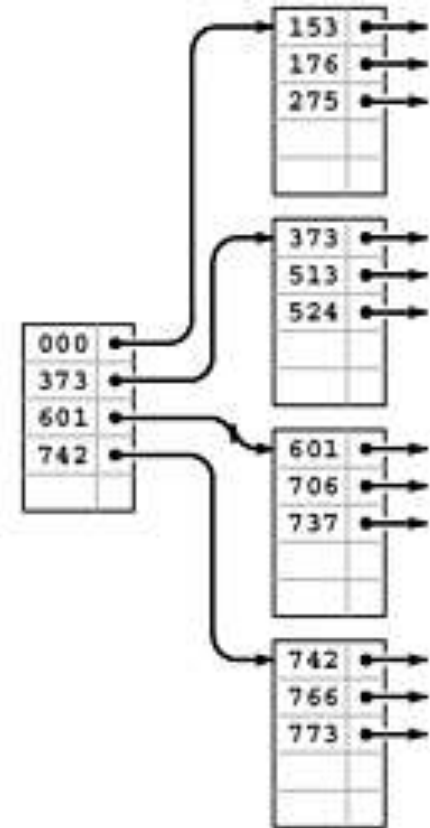
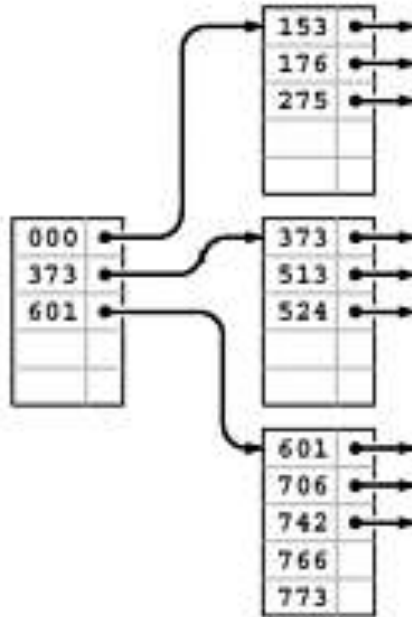
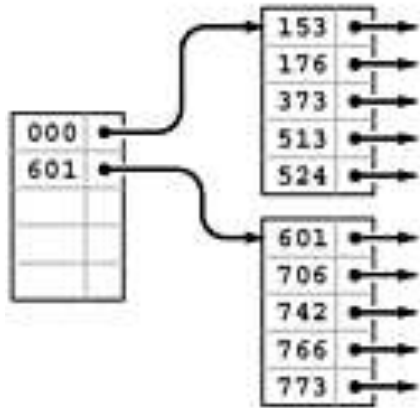


Vantagens da Árvore B+ sobre a B

- ▶ Nó folha e não-folha são do mesmo tamanho
 - ▶ Facilita o gerenciamento do armazenamento para o índice;
- ▶ A remoção é mais simples, pois a entrada a ser removida sempre estará numa folha.



B+ - Aplicações



- Indexação em Banco de Dados





Exercícios



Exercícios

- ▶ Qual a diferença entre uma árvore-B e uma B^* ? Que melhoras a B^* oferece sobre a árvore-B, e que complicações ela introduz?
- ▶ Dada as três variações da árvore B, compare a profundidade de uma em relação a outra.





Fonte dos exemplos e exercícios

- ▶ <http://wiki.icmc.usp.br/images/8/8e/SCC578920131-B.pdf>
- ▶ [http://www.inf.ufrgs.br/~irmmenezes/lib/exe/fetch.php?media=user:trabalho final cpd.pdf](http://www.inf.ufrgs.br/~irmmenezes/lib/exe/fetch.php?media=user:trabalho%20final%20cpd.pdf)
- ▶ <http://homepages.dcc.ufmg.br/~rainerpc/cursos/pa/aulas/10arvoresb+.pdf>
- ▶ <http://www.ic.unicamp.br/~sandro/cursos/mc202/web/documentos/lista2/lista2.html>

