



Estruturas Avançadas de Dados I

Árvores B

1

Árvores B (B-trees) - Introdução

- É uma árvore N-ária, ou seja, uma árvore com um número máximo de subárvores;
- Árvores B (ou B-Trees) foram projetadas para:
 - Manter a árvore balanceada e evitar o desperdício de espaço dentro de um nó à custa de maior complexidade nos algoritmos de inserção e remoção. Feita para gerenciamento de blocos do disco rígido.
- Introduzida por: R. Bayer e E. McCreight em:
- “Organization and Maintenance of Large Ordered Indexes”, Acta Informatica, 1(3), Feb 1972.
- Revista por: D. Comer: “The Ubiquitous B-tree”, ACM Computing Surveys, 11(2), 1979.



4

Árvores B (B-trees) - Características

- Operações de inserção e remoção são **balanceadas** de modo que cada nó tenha uma ocupação mínima e máxima;
- Uma **ocupação mínima** de 50% é garantida em cada nó (exceto a raiz);
- Procura por um registro requer somente **uma busca em profundidade** da raiz até uma folha apropriada;
- Todos os caminhos da raiz até a folha têm **o mesmo comprimento**;
- Páginas das folhas **são ligadas em sequência** através de ponteiros – podem ser percorridas em sequência nas duas direções.



5

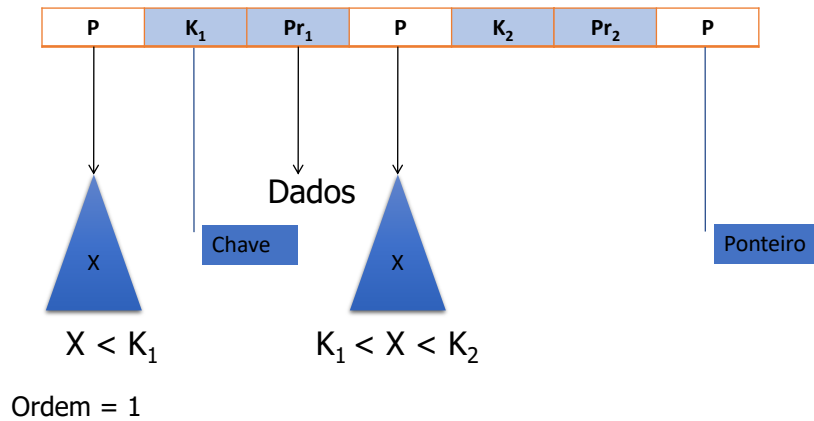
Árvores B (B-trees) - Características

- Os nós são mais comumente chamados de **páginas**, pois, ao contrário das árvores binárias, podemos ter “n” chaves no nó;
- Em uma árvore B de **ordem m**, cada página contém:
 - no mínimo m registros (e m + 1 descendentes);
 - no máximo 2m registros (e 2m + 1 descendentes);
 - exceto a página raiz, que pode conter entre 1 e 2m registros.
- Todas as páginas folha aparecem no mesmo nível.



6

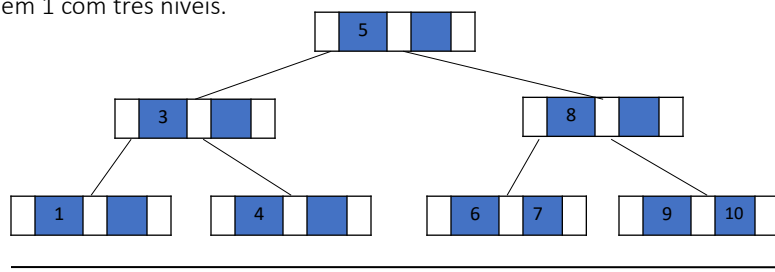
Árvores B (B-trees) – Nó ou Página



7

Árvores B (B-trees) - Exemplo

- Árvore B de ordem 1 com três níveis.



- Todas as páginas contém 1 ou 2 registros.
- A raiz, poderá conter apenas um registro, quando a ordem for maior que 1.
- Os registros aparecem em ordem crescente da esquerda para a direita.
- É uma extensão natural da organização da árvore binária de pesquisa.

8

Árvores B (B-trees) – Inserir

- Primeiro passo: pesquise a chave para ter a certeza de que esta não existe na árvore;
- Segundo passo: busque a posição onde esta será inserida. Se a página não estiver cheia, insira o valor dentro dela (reordenando as suas chaves, caso seja necessário), senão execute uma subdivisão da página da seguinte forma:
 - Verifique se a página pai está cheia, se não:
 - Passe o elemento do meio da página para seu pai.
 - Divida a página em duas páginas iguais.
 - Se a página pai estiver cheia, repita as duas linhas acima recursivamente. Caso todas as páginas-pais estiverem cheias, inclusive a raiz, deve ser criada uma nova página aumentando assim a altura da árvore.
 - Somente após satisfazer todas as divisões necessárias, insira a nova chave.



9

Árvores B (B-trees) – Exemplo ordem = 1

- Inserindo Daniel



10

Árvores B (B-trees) – Exemplo ordem = 1

- Inserindo André



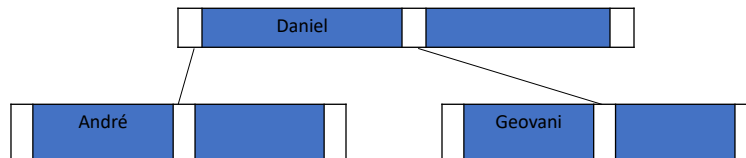
- Ordenar as duas chaves de forma a manter a classificação



11

Árvores B (B-trees) – Exemplo ordem = 1

- Inserindo Geovani



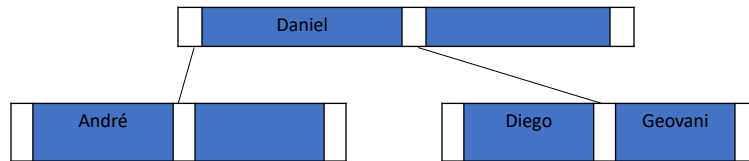
- Ocorrerá overflow, ou seja, a página está cheia
- A chave do meio é Daniel, que será promovida como pai da página atual. Ocorrerá a divisão (cisão) da página



12

Árvores B (B-trees) – Exemplo ordem = 1

- Inserindo Diego



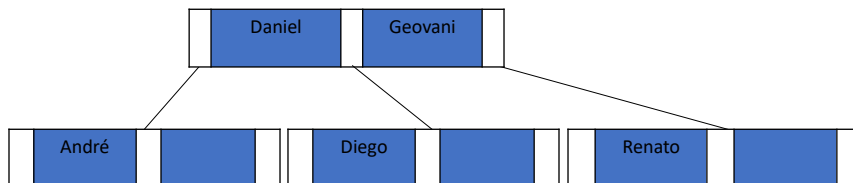
- Navegar na árvore até achar a página folha (página com o Geovani);
- Inserir a chave reorganizando a página.



13

Árvores B (B-trees) – Exemplo ordem = 1

- Inserindo Renato



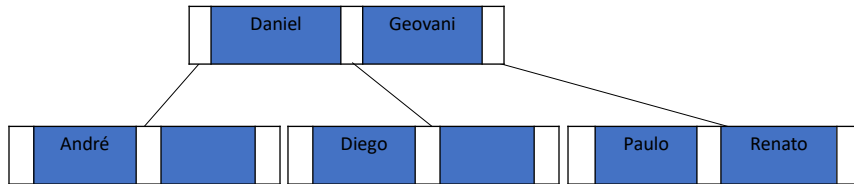
- Navegar até a página folha (página com o Diego e Geovani);
- Como a página está cheia é necessário fazer a divisão;
- A chave do meio é promovida para a raiz;
- A página original é dividida. Como há espaço na raiz, ela não terá divisão.



14

Árvores B (B-trees) – Exemplo ordem = 1

- Inserindo Paulo



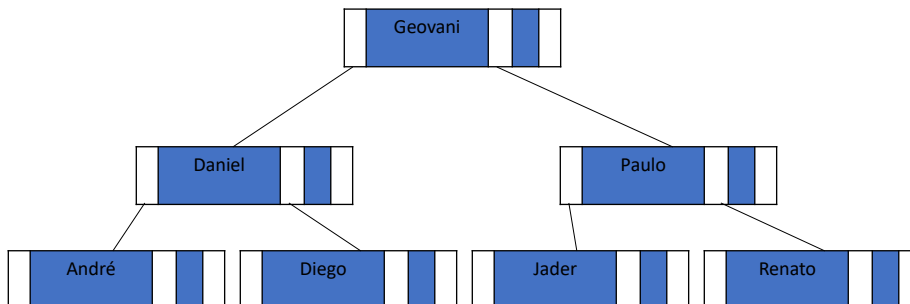
- Navegar até a página folha (página com o Renato);
- Como a página não está cheia, apenas inserir.



15

Árvores B (B-trees) – Exemplo ordem = 1

- Inserindo Jader



- Navegar até a página folha (página com o Paulo e Renato);
- A página está cheia, então, é dividido;
- A chave Paulo é movida para a página raiz;
- A página raiz está cheia, então, é dividido e uma nova página criada.



16

Árvores B (B-trees) – Considerações

- São necessários 3 acessos para encontrar o Renato
- Em relação à utilização:

$$\frac{\text{Registros Utilizados}}{\text{Registros Totais}} = \frac{7}{14} = 50\%$$

- Como a Árvore B é balanceada, no pior caso, o número total de acessos é igual a altura da árvore para a ordem “m”:

$$h \leq \log_{m+1} \frac{n+1}{2} + 1$$

- Onde “n” é o número de registros armazenados.
- Portanto, no pior caso, a complexidade é $O(\log_m n)$.



17

Árvores B (B-trees) – Complexidade

- Ordem $m = 1$
- $n = 7$

$$h \leq \log_{1+1} \frac{7+1}{2} + 1$$

$$h \leq \sim 2,3$$

- Ou seja, no máximo 3 acessos.

- Ordem $m = 256$
- $n = 1.000.000$

$$h \leq \log_{256+1} \frac{1.000.000+1}{2} + 1$$

$$h \leq \sim 2,3$$

- Se fizéssemos com um ABP teríamos: $\sim 19,93$ acessos $\log_2(1.000.000)$



18

Árvores B (B-trees) – Complexidade

- Na inserção, cada cisão (split ou divisão) opera sobre um número fixo de páginas, portanto, sua complexidade é de $O(1)$ acessos. No pior caso, overflows se propagarão até a raiz, resultando em complexidade $O(\log_m n)$ splits com $O(1)$ acessos cada;
- Na remoção, cada concatenação opera sobre um número fixo de páginas, logo, sua complexidade é de $O(1)$ acessos. No pior caso, underflows se propagarão até a raiz, resultando em complexidade $O(\log_m n)$ concatenações com $O(1)$ acessos cada;
- Na busca, o pior caso será o maior nível (ou a altura) da árvore, que, neste caso é $O(\log_m n)$ acessos.



19

Árvores B (B-trees) – Excluir

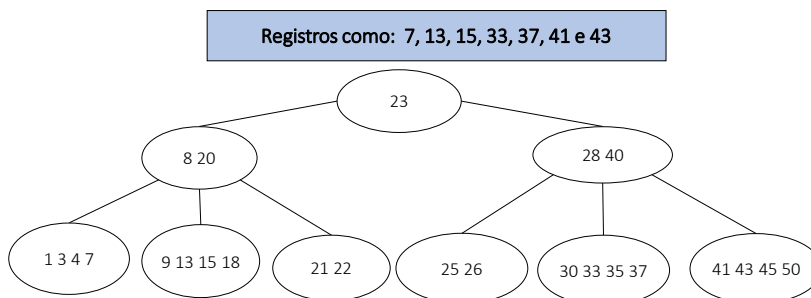
- Primeiro, é preciso localizar a página apropriada do registro a ser excluído
- Duas possibilidades:
 - Caso 1: quando o registro se encontra em uma página folha.
 - 1.1: A folha possui mais que m registros
 - 1.2: A folha possui apenas m registros e o irmão possui $m+1$ registros
 - 1.3: A folha e seus irmãos possuem apenas m registros
 - Caso 2: quando o registro não se encontra em uma folha. Nesse caso, o registro a ser retirado deve ser primeiro substituído por um outro para depois ser excluído.
- Em ambos os casos deve-se verificar se a retirada não afeta as propriedades básicas de uma árvore B



20

Árvores B (B-trees) – Excluir

Caso 1.1: quando o registro se encontra em uma página folha e a folha possui mais que m registros. Ordem = 2.



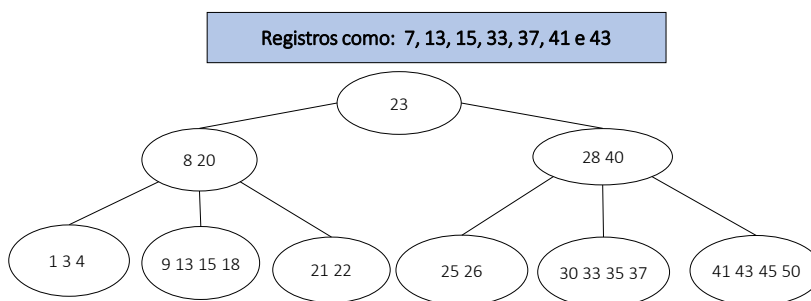
Retirada simples do elemento da folha



21

Árvores B (B-trees) – Excluir

Caso 1.1: quando o registro se encontra em uma página folha e a folha possui mais que m registros. Ordem = 2.



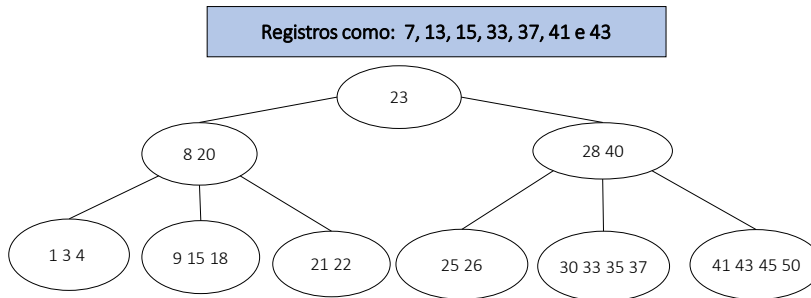
Retirada simples do elemento da folha



22

Árvores B (B-trees) – Excluir

Caso 1.1: quando o registro se encontra em uma página folha e a folha possui mais que m registros. Ordem = 2.



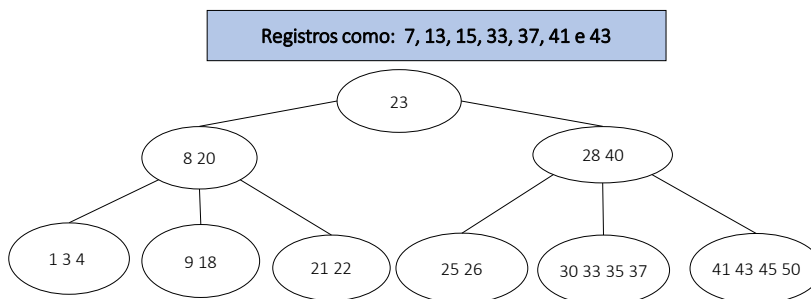
Retirada simples do elemento da folha



23

Árvores B (B-trees) – Excluir

Caso 1.1: quando o registro se encontra em uma página folha e a folha possui mais que m registros. Ordem = 2.



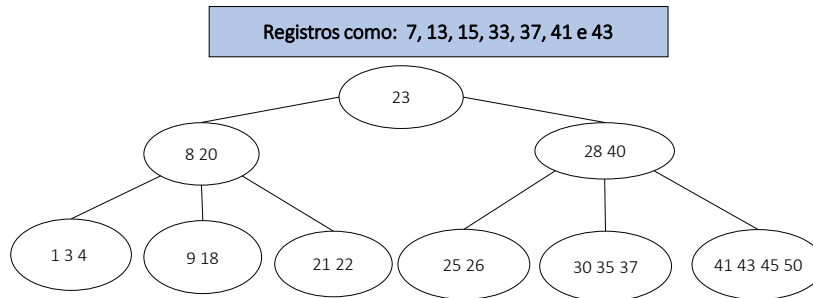
Retirada simples do elemento da folha



24

Árvores B (B-trees) – Excluir

Caso 1.1: quando o registro se encontra em uma página folha e a folha possui mais que m registros. Ordem = 2.



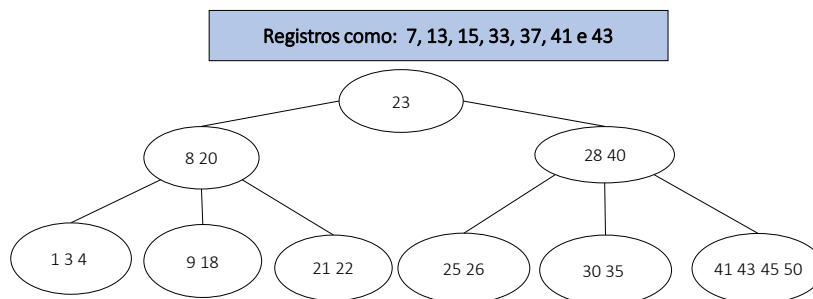
Retirada simples do elemento da folha



25

Árvores B (B-trees) – Excluir

Caso 1.1: quando o registro se encontra em uma página folha e a folha possui mais que m registros. Ordem = 2.



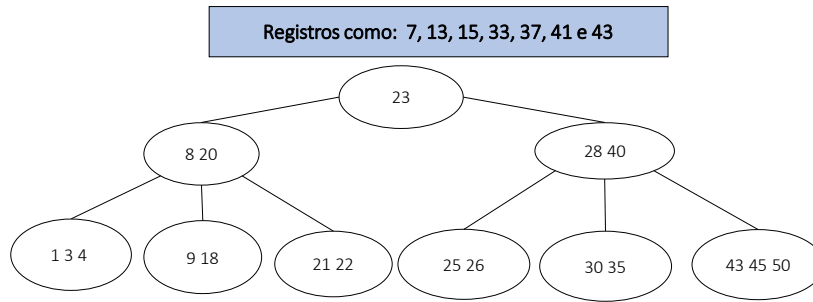
Retirada simples do elemento da folha



26

Árvores B (B-trees) – Excluir

Caso 1.1: quando o registro se encontra em uma página folha e a folha possui mais que m registros. Ordem = 2.



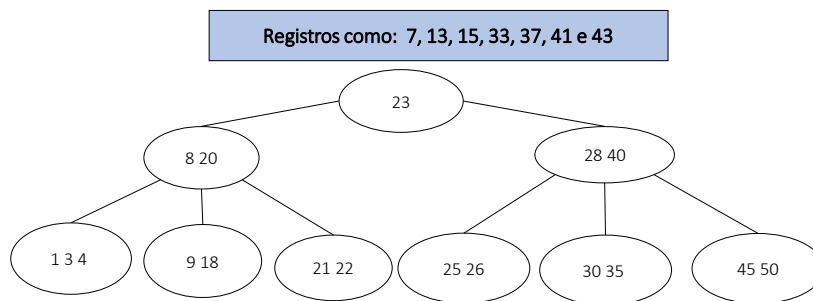
Retirada simples do elemento da folha



27

Árvores B (B-trees) – Excluir

Caso 1.1: quando o registro se encontra em uma página folha e a folha possui mais que m registros. Ordem = 2.



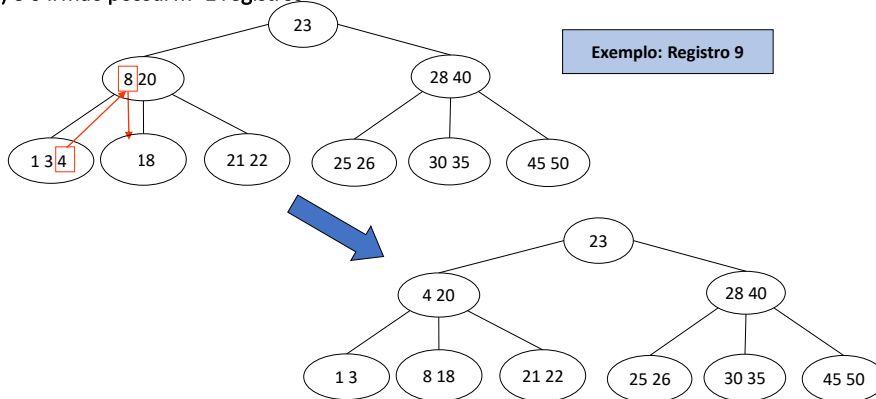
Retirada simples do elemento da folha



28

Árvores B (B-trees) – Excluir

Caso 1.2: quando o registro se encontra em uma página folha e a folha possui m registros (mínimo possível) e o irmão possui $m+1$ registros



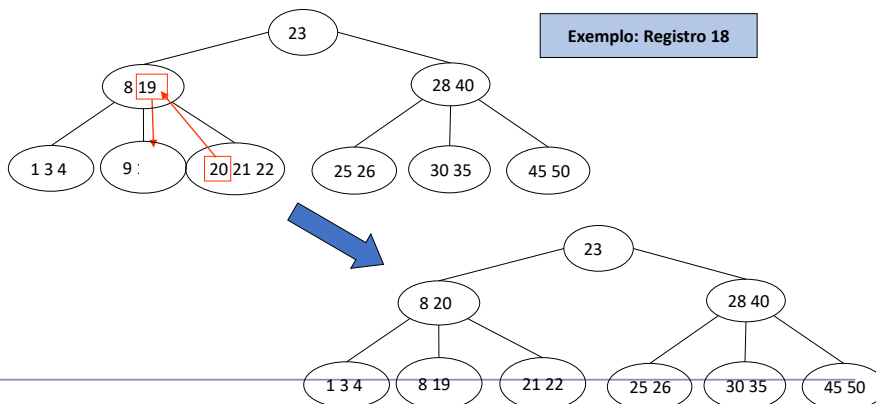
A chave k do pai que separa os irmãos pode ser incluída na página X e a última ou primeira chave do irmão (última se o irmão for da esquerda e primeira se o irmão for da direita) pode ser inserida no pai no lugar de k .



29

Árvores B (B-trees) – Excluir

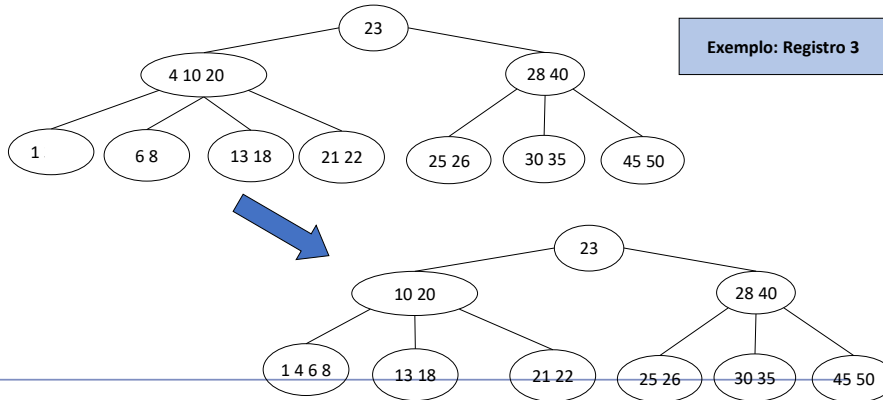
Caso 1.2: quando o registro se encontra em uma página folha; a folha possui m registro (mínimo possível) e o irmão possui $m+1$ registros



30

Árvores B (B-trees) – Excluir

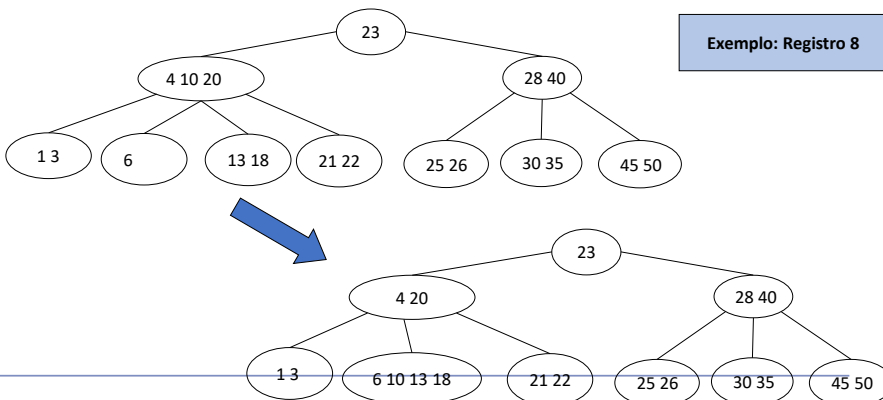
Caso 1.3.1: quando o registro se encontra em uma página folha; a folha e seus irmãos possuem m registros (mínimo possível) e o pai pode emprestar.



31

Árvores B (B-trees) – Excluir

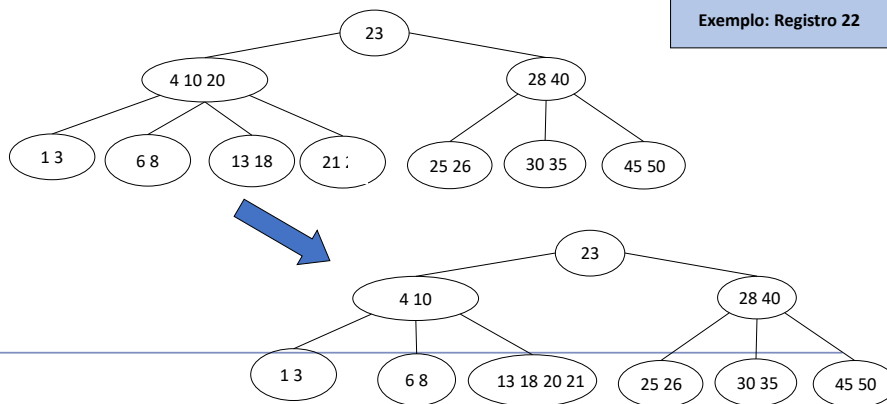
Caso 1.3.1: quando o registro se encontra em uma página folha; a folha e seus irmãos possuem m registros (mínimo possível) e o pai pode emprestar.



32

Árvores B (B-trees) – Excluir

Caso 1.3.1: quando o registro se encontra em uma página folha, a folha e seus irmãos possuem m registros (mínimo possível) e o pai pode emprestar.

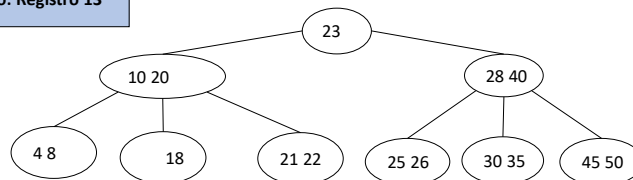


33

Árvores B (B-trees) – Excluir

Caso 1.3.2: quando o registro se encontra em uma página folha; a folha e seus irmãos possuem m registros (mínimo possível) e o pai não pode emprestar.

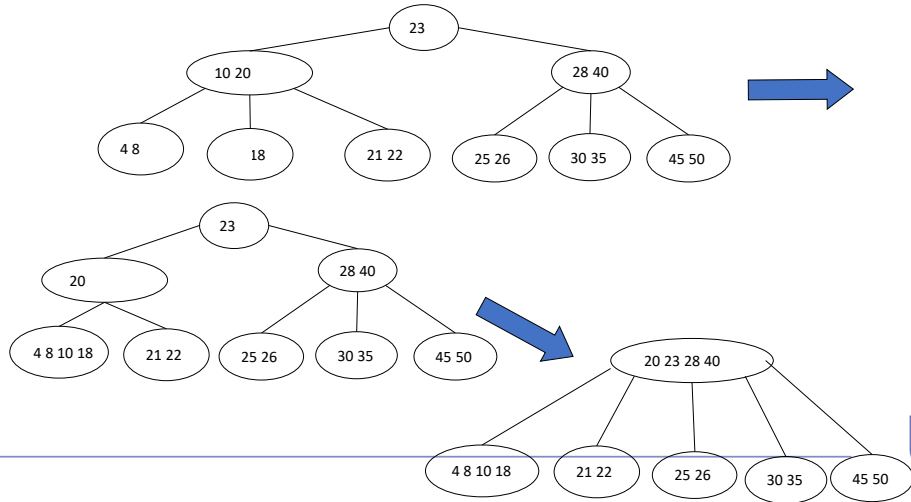
Exemplo: Registro 13



Se os dois irmãos de X e o pai contiverem exatamente m registros (ocupação mínima), nenhum registro poderá ser emprestado. Neste caso, a página X e um de seus irmãos serão concatenados (underflow) em uma única página que também contém a chave separadora do pai, e o procedimento é feito recursivamente até que as páginas contenham a quantidade mínima de registros.

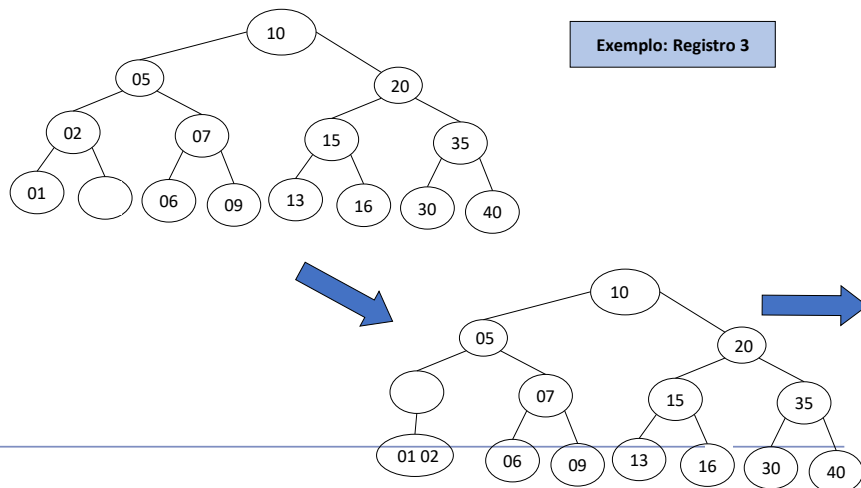
34

Árvores B (B-trees) – Excluir



35

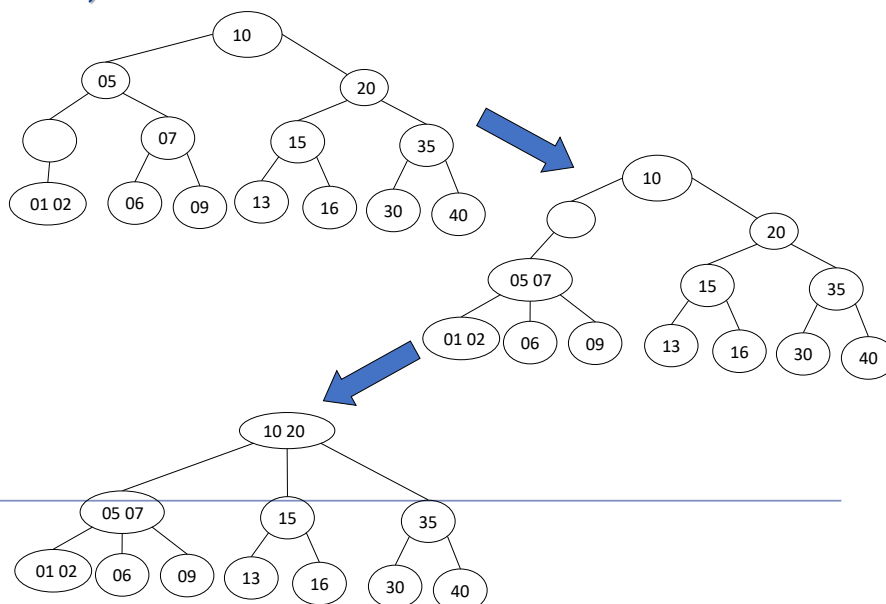
Árvores B (B-trees) – Excluir



Exemplo: Registro 3

36

Árvores B (B-trees) – Excluir



37

Árvores B (B-trees) – Excluir

Caso 2: quando o registro não se encontra em uma folha.

Mesmo procedimento de remoção em árvores binárias ou AVL

Substituição pela maior chave da sub-árvore à esquerda;

Substituição pela menor chave da sub-árvore à direita.

Caso 2.1: a sub-árvore vizinha possui registros para emprestar;

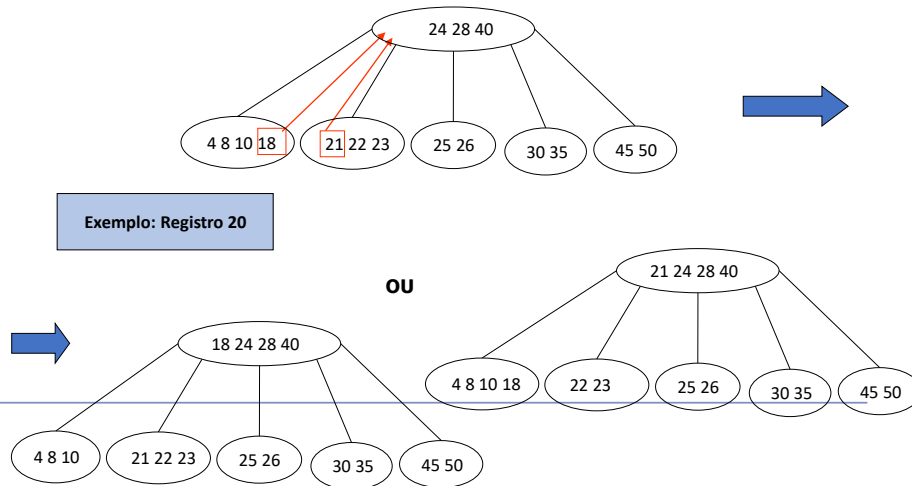
Caso 2.2: a sub-árvore vizinha não possui registros para emprestar.



38

Árvores B (B-trees) – Excluir

Caso 2.1: a sub-árvore vizinha possui registros para emprestar.



39

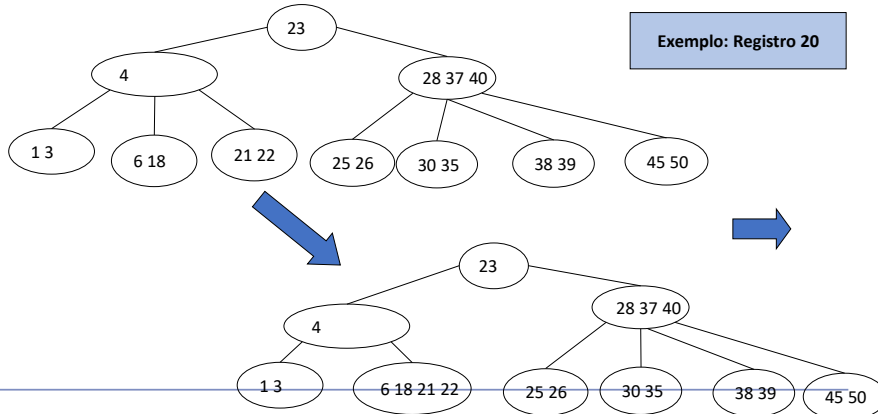
Árvores B (B-trees) – Excluir

- Caso 2: quando o registro não se encontra em uma folha.
- Possui dois casos:
 - Caso 2.2.1: O irmão possui registros para emprestar
 - Caso 2.2.2: O irmão não possui registros para emprestar

40

Árvores B (B-trees) – Excluir

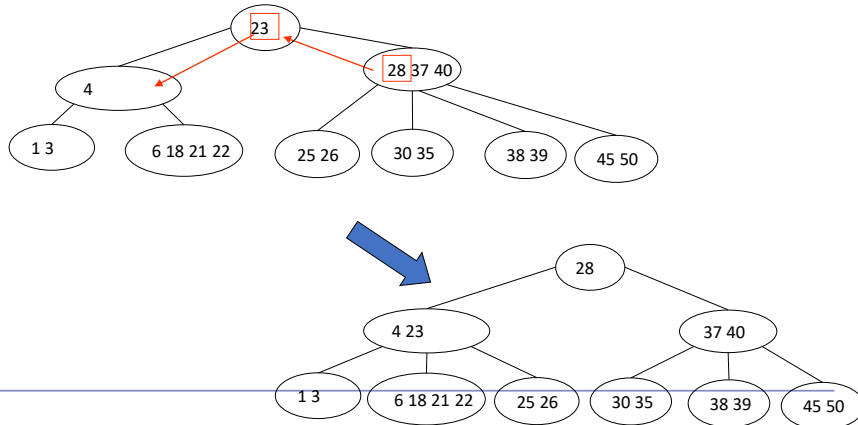
- Caso 2.2.1: a sub-árvore vizinha NÃO possui registros para emprestar e o irmão possui registros para emprestar.



41

Árvores B (B-trees) – Excluir

- Caso 2.2.1: a sub-árvore vizinha NÃO possui registros para emprestar e o irmão possui registros para emprestar.

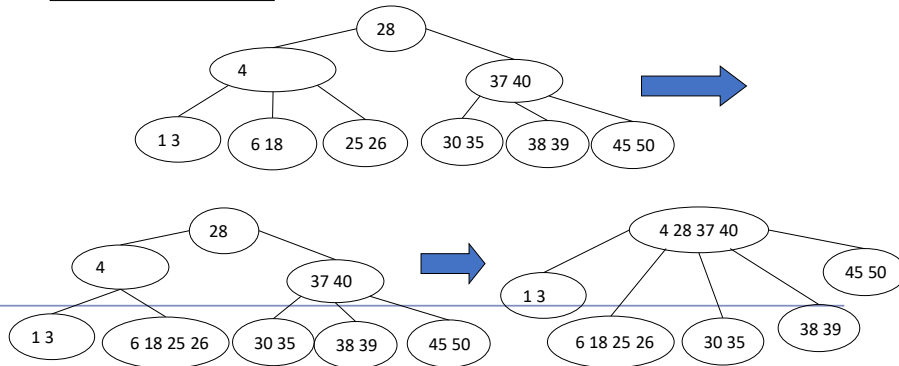


42

Árvores B (B-trees) – Excluir

- Caso 2.2.2: a sub-árvore vizinha NÃO possui registros para emprestar e o irmão também NÃO possui registros para emprestar.

Exemplo: Registro 23



43

Exercícios

6.1 - Mostre passo-a-passo a inserção das chaves de “A” a “G” em uma árvore B. Ordem = 1.

6.2 - Mostre a árvore em cada uma das três fases. Ordem = 2.

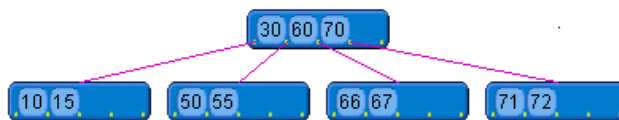
Inserir os valores: 20, 10, 40, 50, 30.

Inserir os valores: 55, 3, 11, 4, 28, 36, 33, 52, 17, 25 e 13

Inserir os valores: 45, 9, 43, 8, 48.

6.3 – Excluir as seguintes chaves: 30, 72, 55, 66, 60, 70, 10

Ordem = 2



45

Exercícios

6.4 – Em uma árvore B de ordem $m=1024$, defina:

- Quantas páginas filhas terá uma página não folha e não raiz?
- Quantos acessos teríamos no pior caso, considerando $n = 512$? Em uma ABP quantos acessos teríamos neste mesmo cenário?

6.5 – Apresente um exemplo de divisão (split ou cisão ou overflow) em uma árvore B, bem como o exemplo de concatenação (ou underflow). Ordem = 1.



46

Referências Bibliográficas

- CORMEN, Thomas H. et al. **Introduction to algorithms**. 3. ed. Cambridge: MIT, 2009. xix. 1292 p.
- CAMPELLO, R. J. G. B. **Árvores B – Parte I e II**. Apresentação PPT.



47

Prof. Márcio Garcia Martins
marciog@unisinos.br

Para anotar: *ao enviar e-mail sempre coloque o
seguinte prefixo no assunto*
[EADI-ano-semester] – Nome do aluno

