

Algoritmos e Estruturas de Dados III

3º Período Engenharia da Computação

Prof. Edwaldo Soares Rodrigues

Email: edwaldoroadsf1@yahoo.com.br

Árvores B

- São árvores de pesquisa balanceadas projetadas para funcionar bem em discos magnéticos ou outros dispositivos de armazenamento secundário (Cormen, T.);
- Muitos SGBDs usam árvores B ou variações de árvores B para armazenar informações;
- Método genérico para o armazenamento e a recuperação de dados:
 - Voltado para arquivos volumosos;
 - Proporciona rápido acesso aos dados;
 - Possui custo mínimo de overhead;

Árvores B

- Características:
 - Índice:
 - Extremamente volumoso;
 - *Buffer-pool* pequeno:
 - Apenas uma parcela do índice pode ser carregada em memória principal;
 - Operações baseadas em disco;
 - Normalmente um nó da árvore B é tão grande quanto uma página de disco inteira;

Árvores B

- Desempenho:
 - Uma árvore B com n itens e t sendo o tamanho de um bloco, tem complexidade de E/S:
 - $O(\log_t n)$ para operações de pesquisa/atualização e usa $O(n/t)$ blocos;
 - O número de acessos ao disco exigidos para a maioria das operações em uma árvore B é proporcional a sua altura (Cormen, T.);
 - Os descendentes são divididos o mais uniformemente possível entre as páginas velha e nova;

Árvores B

- **Pesquisa em memória secundária:** arquivos contém mais registros do que a memória interna pode armazenar;
- Custo para acessar um registro é algumas ordens de grandeza maior do que o custo de processamento na memória primária;
- **Medida de complexidade:** custo de transferir dados entre a memória principal e secundária (minimizar o número de transferências);
- **Memórias secundárias:** apenas um registro pode ser acessado em um dado momento (acesso sequencial);

Árvores B

- **Memórias primárias:** acesso a qualquer registro de um arquivo a um custo uniforme (acesso direto);
- O aspecto sistema de computação é importante;
- As características da arquitetura e do sistema operacional da máquina tornam os métodos de pesquisa dependentes de parâmetros que afetam o desempenho;

Árvores B

- Proposto em 1972 por Bayer e McCreight, desenvolvido no Laboratório de Pesquisas Científicas Boeing;
- A origem do nome nunca foi explicada pelos autores;
 - “B” de Bayer ou “B” de Boeing;

Árvores B

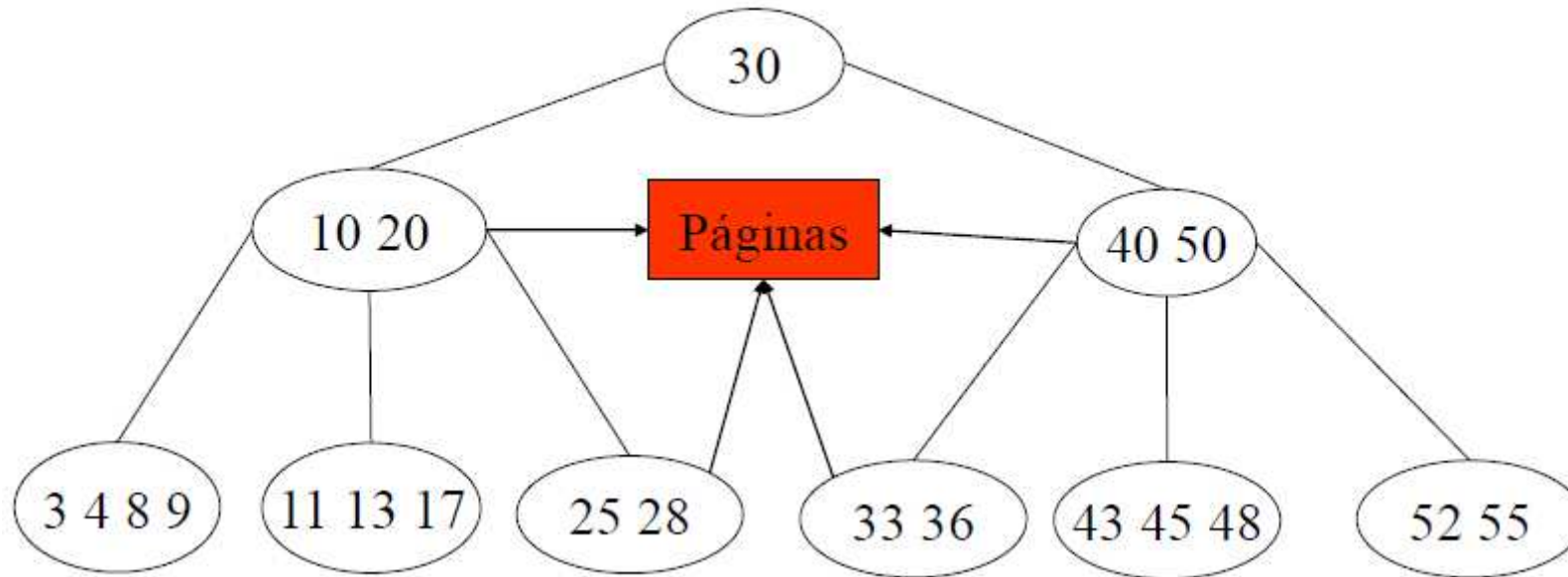
- Definições:
 - Árvores binárias de pesquisa tem no máximo 2 nós;
 - Quando existem 2 ou mais nós, passam a ser chamadas de n-árias;
 - Nesses casos, os nós são mais comumente chamados de páginas;
 - Os registros de uma árvore B também são armazenados em ordem;

Árvores B

- Supondo uma árvore B n-ária:
 - Em uma árvore B de ordem m tem-se:
 - Cada página contém:
 - Registros: no mínimo m e no máximo $2m$ registros, exceto a raiz, que pode conter entre 1 e $2m$ registros;
 - Descendentes: no mínimo $m+1$ descendentes e no máximo $2m+1$ descendentes;
 - Todos os nós folhas encontram-se no mesmo nível;

Árvores B

- Exemplo de uma árvore B de ordem 2 com 3 níveis:

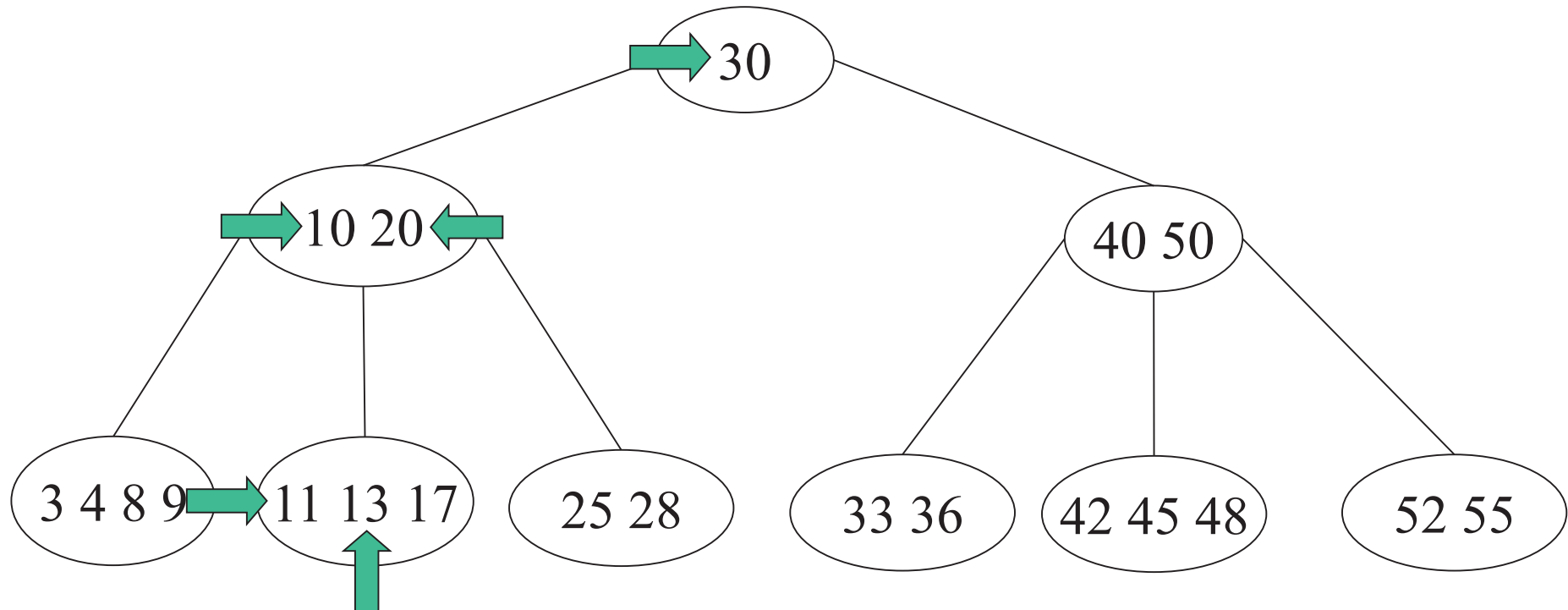


Árvores B

- Algumas das operações que podem ser feitas em árvores B são:
 - Inicializar;
 - Pesquisar;
 - Inserir;
 - Retirar;

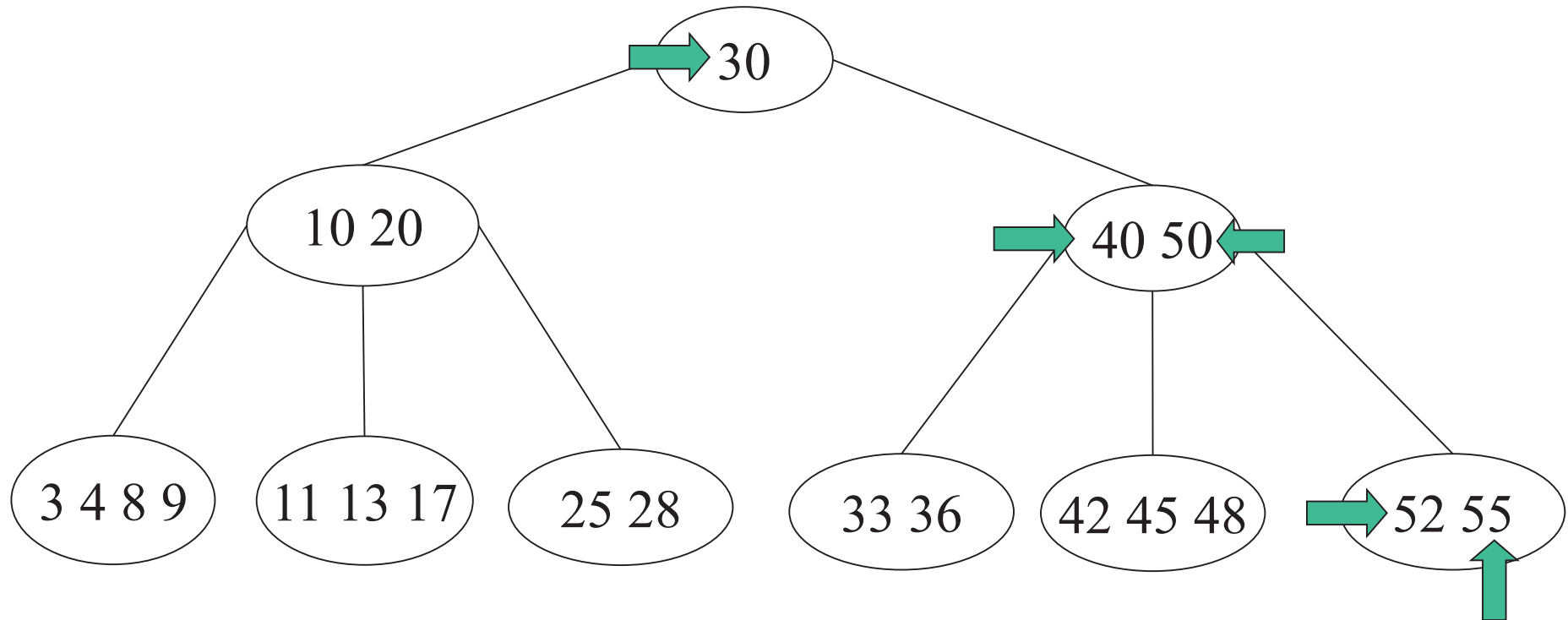
Pesquisa em Árvore B

Procedimento para pesquisa do número 13



Pesquisa em Árvore B

Procedimento para pesquisa do número 55



Inserção

- **Primeiro, é preciso localizar a página apropriada em que o novo registro deve ser inserido**
- **Duas possibilidades:**
 - **Caso 1: o registro encontra seu lugar em uma página com menos de $2m$ registros. Nesse caso, o processo de inserção fica limitado àquela página**
 - **Caso 2: O registro precisa ser inserido em uma página já cheia (com $2m$ registros). Nesse caso, o processo de inserção pode levar à criação de uma nova página.**

Inserção de Registros

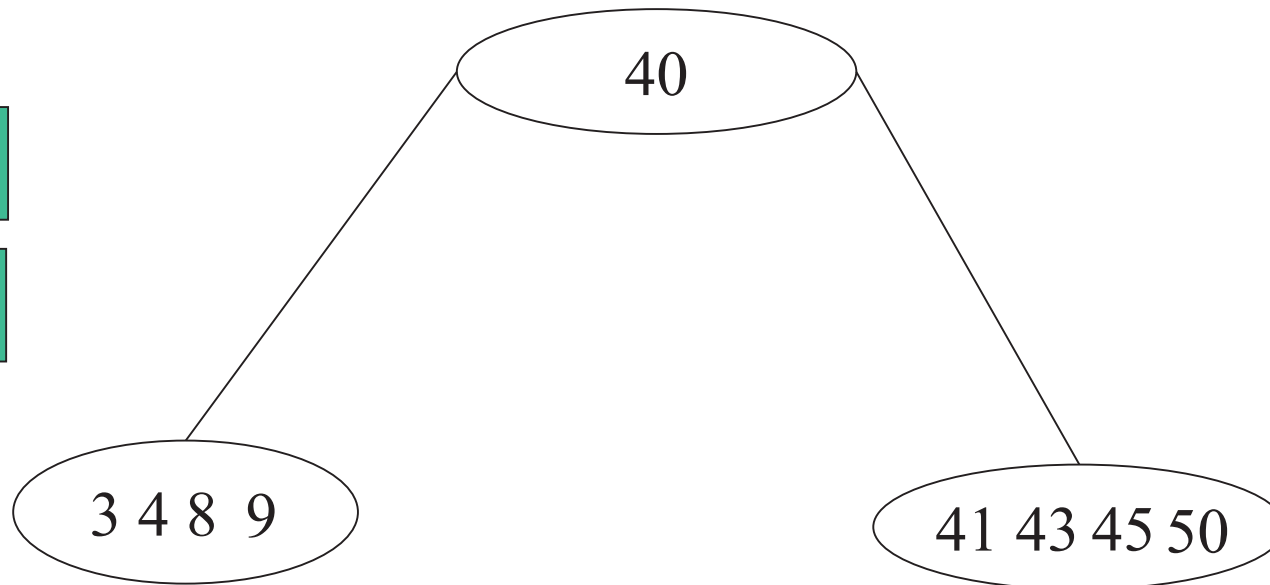
Caso 1: o registro encontra seu lugar em uma página com menos de 2m registros

Ordem 2

Registro 9

Registro 45

Registro 50

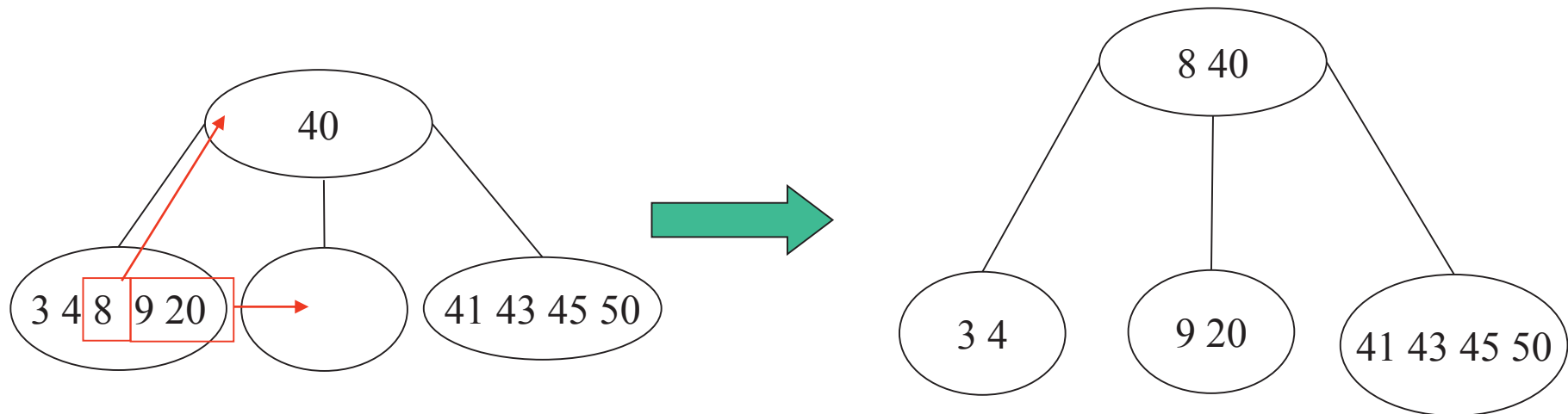
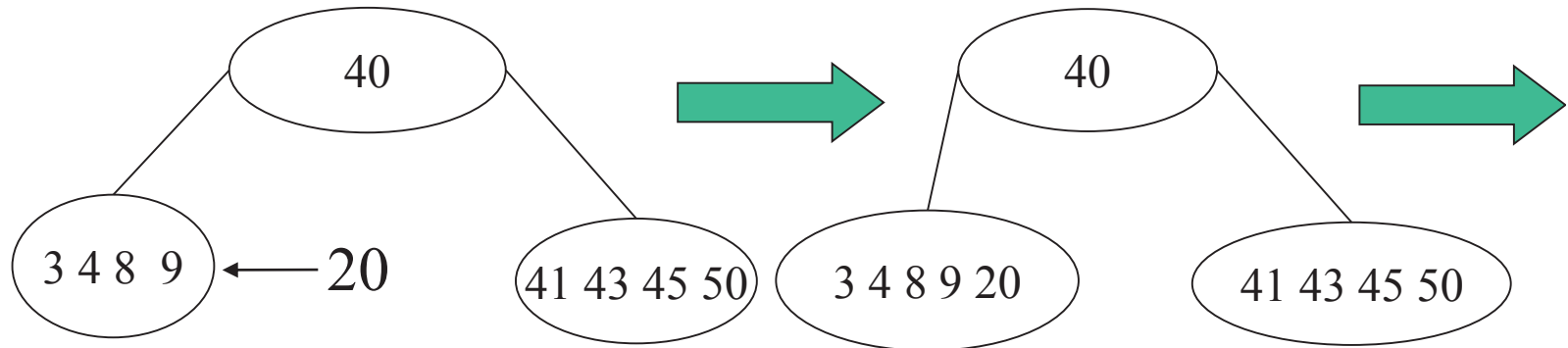


Inserção de Registros

Caso 2: O registro precisa ser inserido em uma página já cheia (com $2m$ registros), e a página pai tem menos que $2m$ registros e $2m+1$ páginas

Ordem: 2

Registro 20

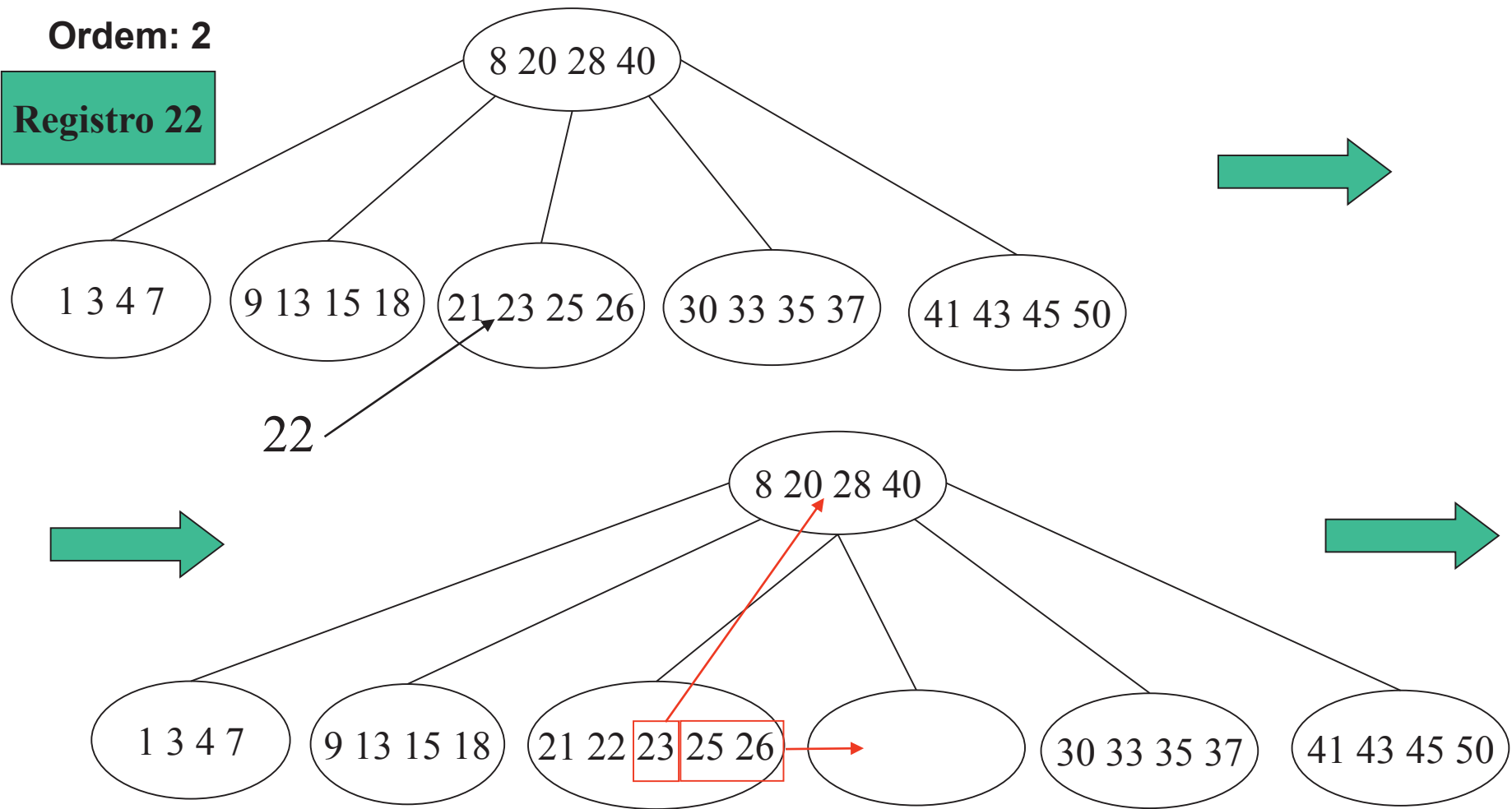


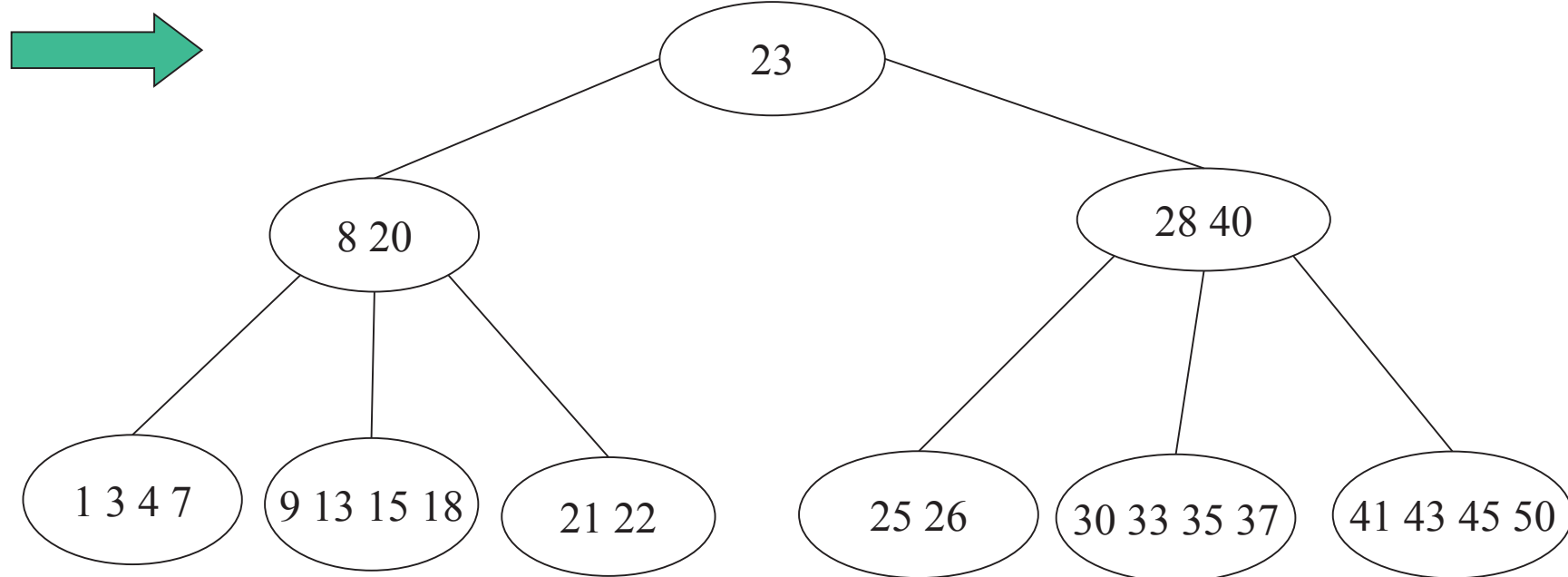
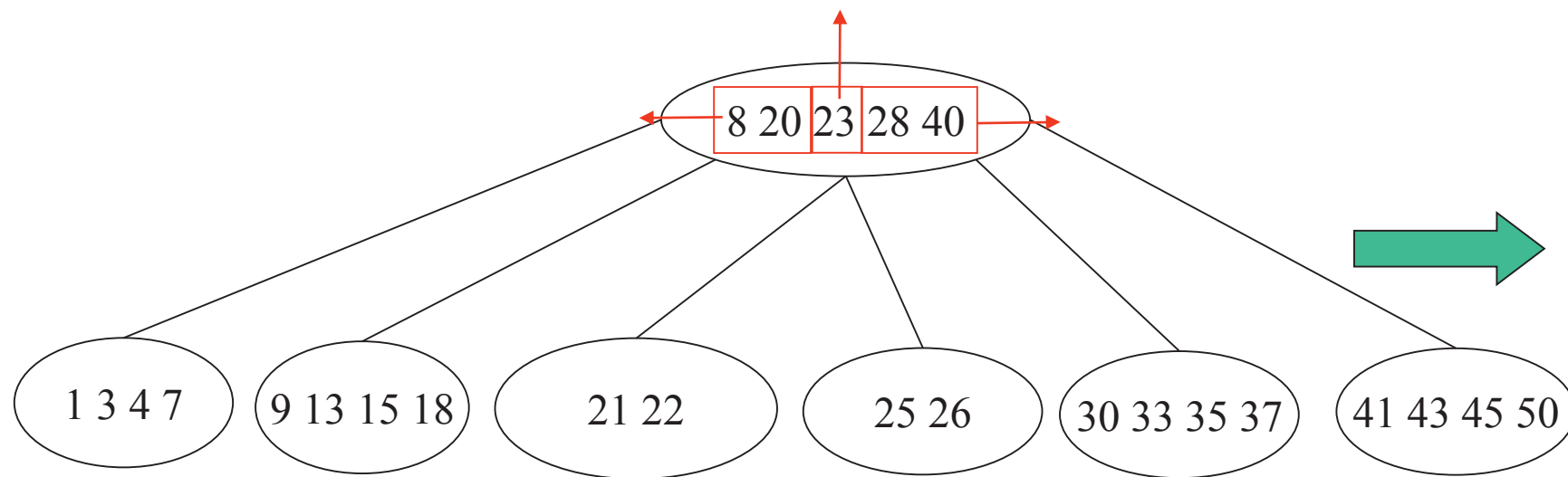
Inserção de Registros

Caso 2: O registro precisa ser inserido em uma página já cheia (com $2m$ registros), e a página pai já tem $2m$ registros e $2m+1$ páginas

Ordem: 2

Registro 22





Remoção

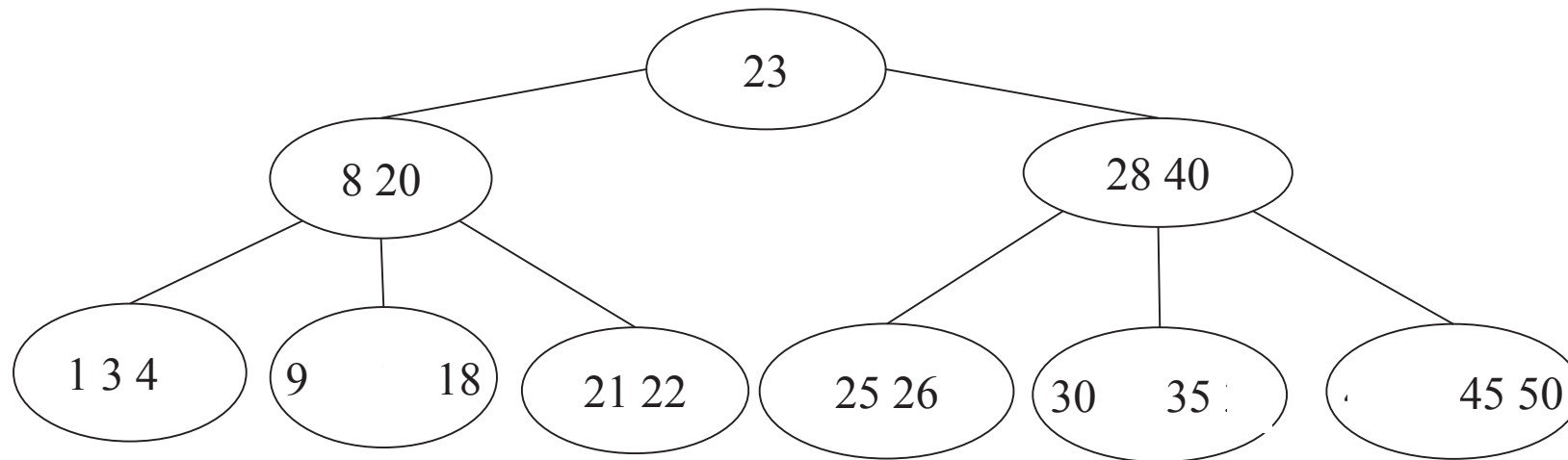
- **Primeiro, é preciso localizar a página apropriada do registro a ser excluído**
- **Duas possibilidades:**
 - **Caso 1: quando o registro se encontra em uma página folha.**
 - 1.1: A folha possui mais que m registros
 - 1.2: A folha possui apenas m registro e o irmão possui $m+1$ registros
 - 1.3: 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**

Remoção de Registros

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

Ordem 2

Registros 7, 13, 15, 33, 37, 41 e 43

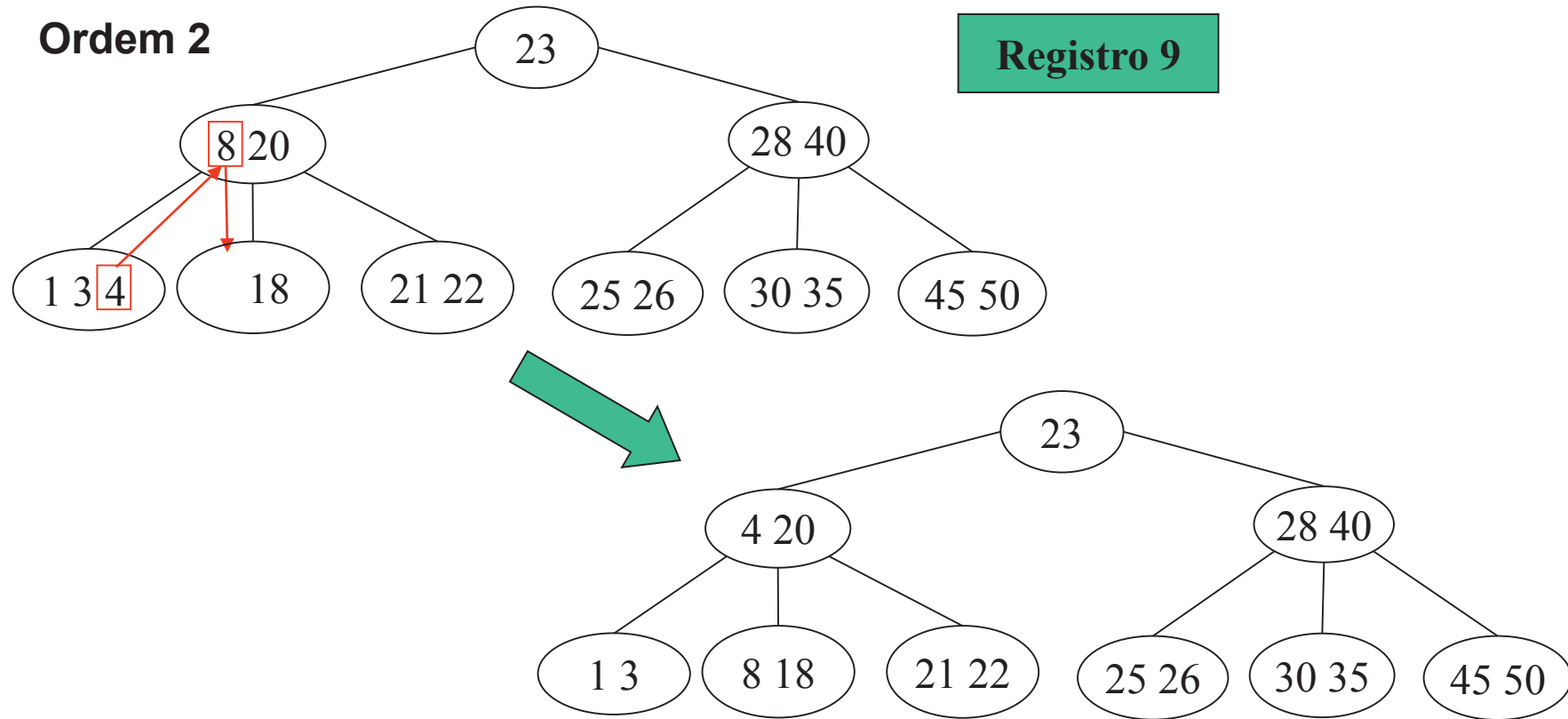


Retirada simples do elemento da folha

Remoção de Registros

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

Ordem 2

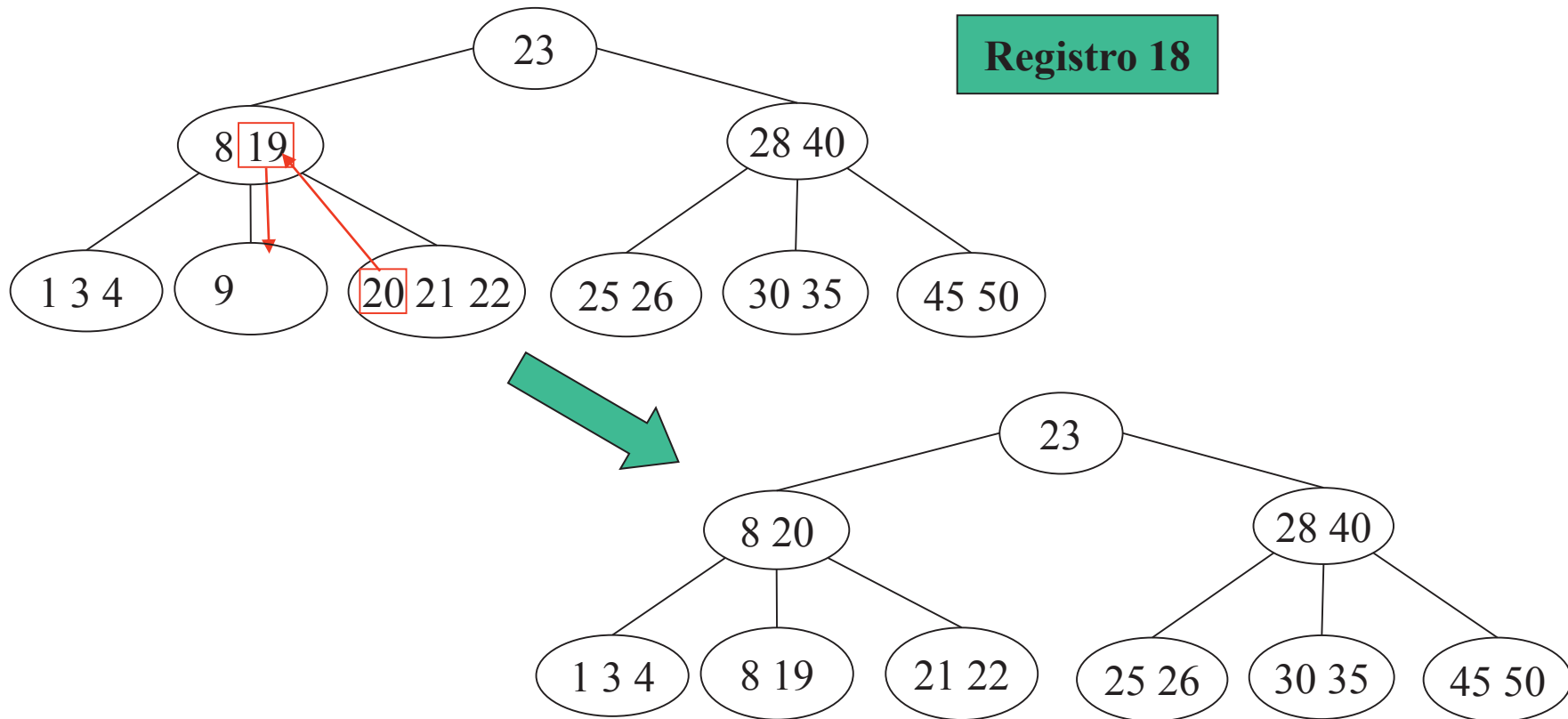


A chave k do pai que separa os irmãos pode ser incluída no nó X e a última ou primeira chave do irmão (última se o irmão for da esquerda e primeira se o irmão for da direita) pode ser inserida no pai no lugar de k .

Remoção de Registros

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

Ordem 2



Remoção de Registros

Caso 1.3: quando o registro se encontra em uma página folha, a folha e seus irmãos possuem m registros (mínimo possível). Se sub-dividem em dois casos:

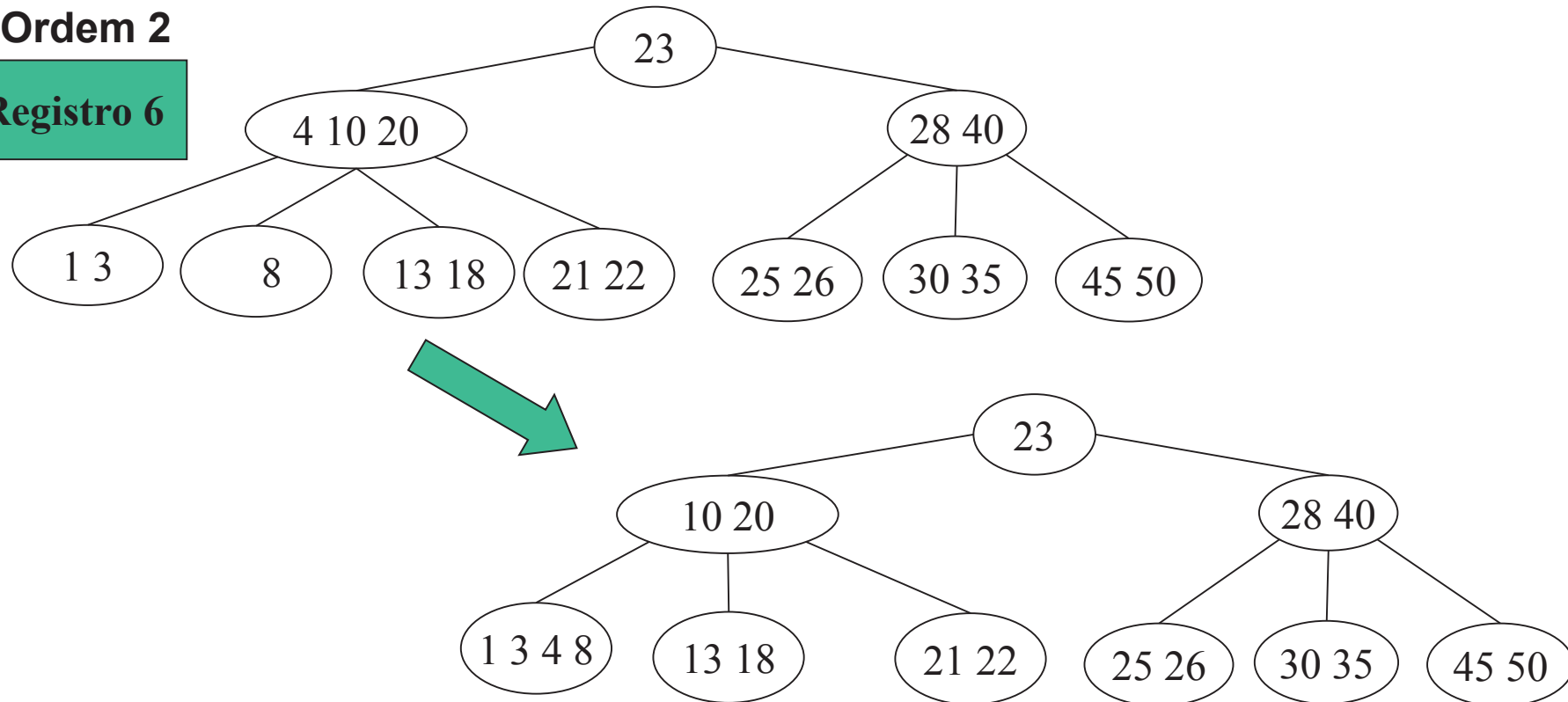
- **Caso 1.3.1: O pai pode emprestar registros**
- **Caso 1.3.2: O pai não pode emprestar registros**

Remoção de Registros

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

Ordem 2

Registro 6



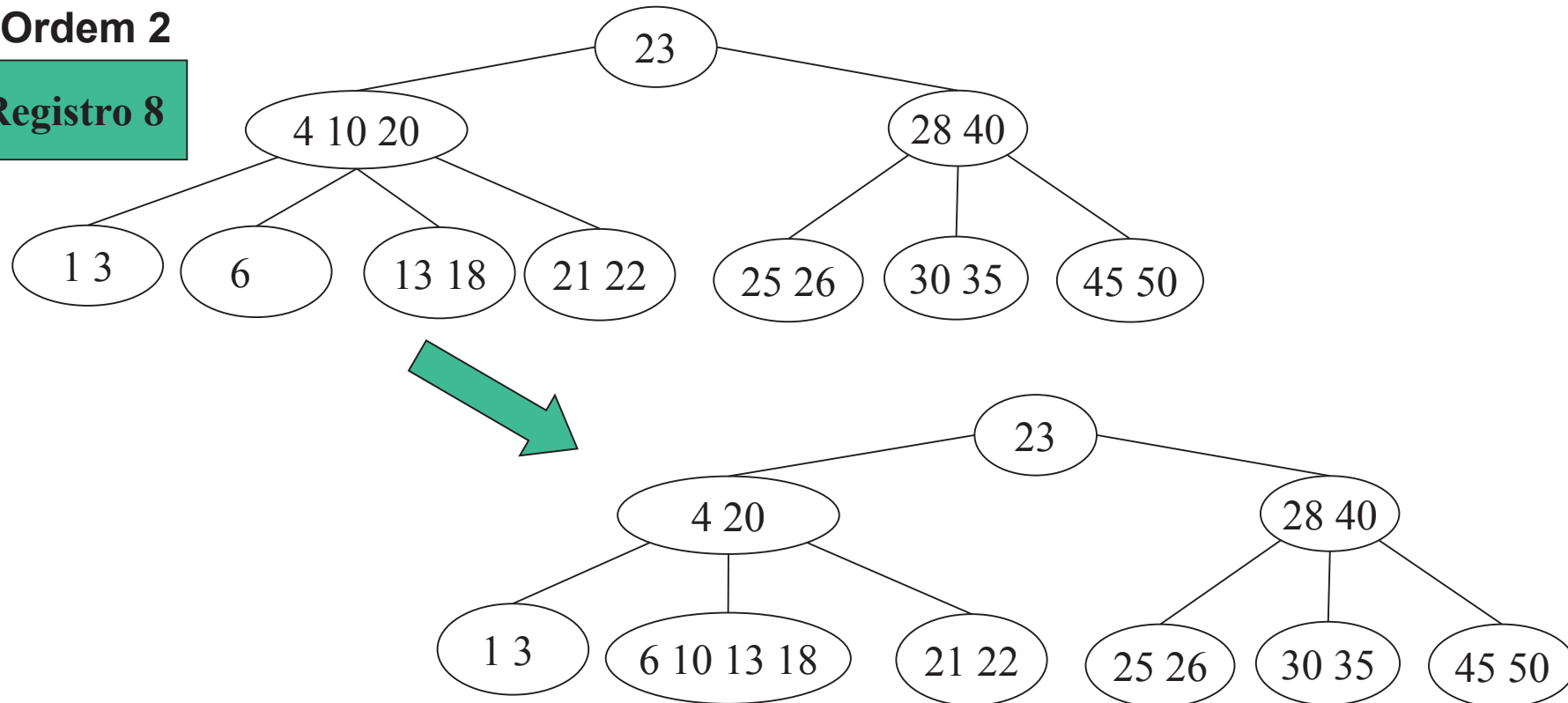
Se os dois irmãos de X contiverem exatamente m registros (ocupação mínima), nenhum registro poderá ser emprestado. Neste caso, o nó X e um de seus irmãos (à esquerda ou direita) são concatenados em um único nó, que também contém a chave separadora do pai.

Remoção de Registros

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

Ordem 2

Registro 8

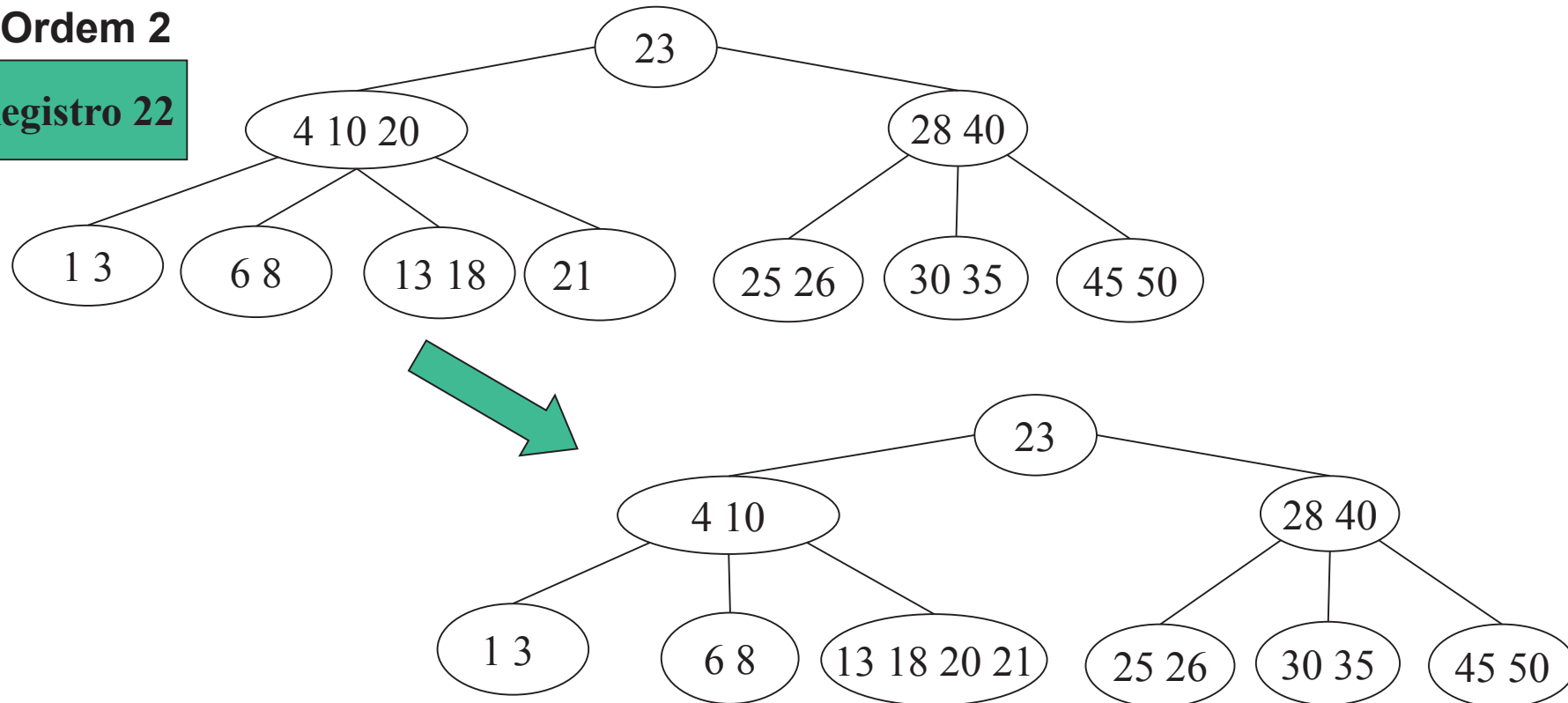


Remoção de Registros

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

Ordem 2

Registro 22

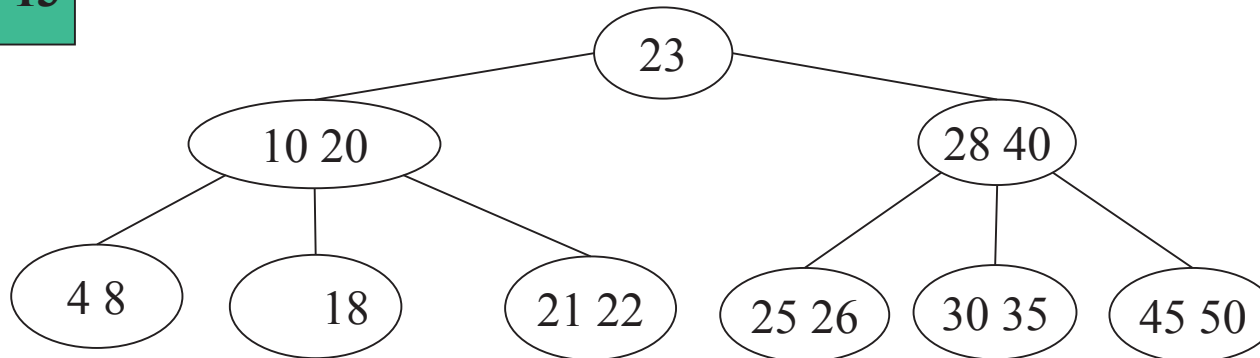


Remoção de Registros

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

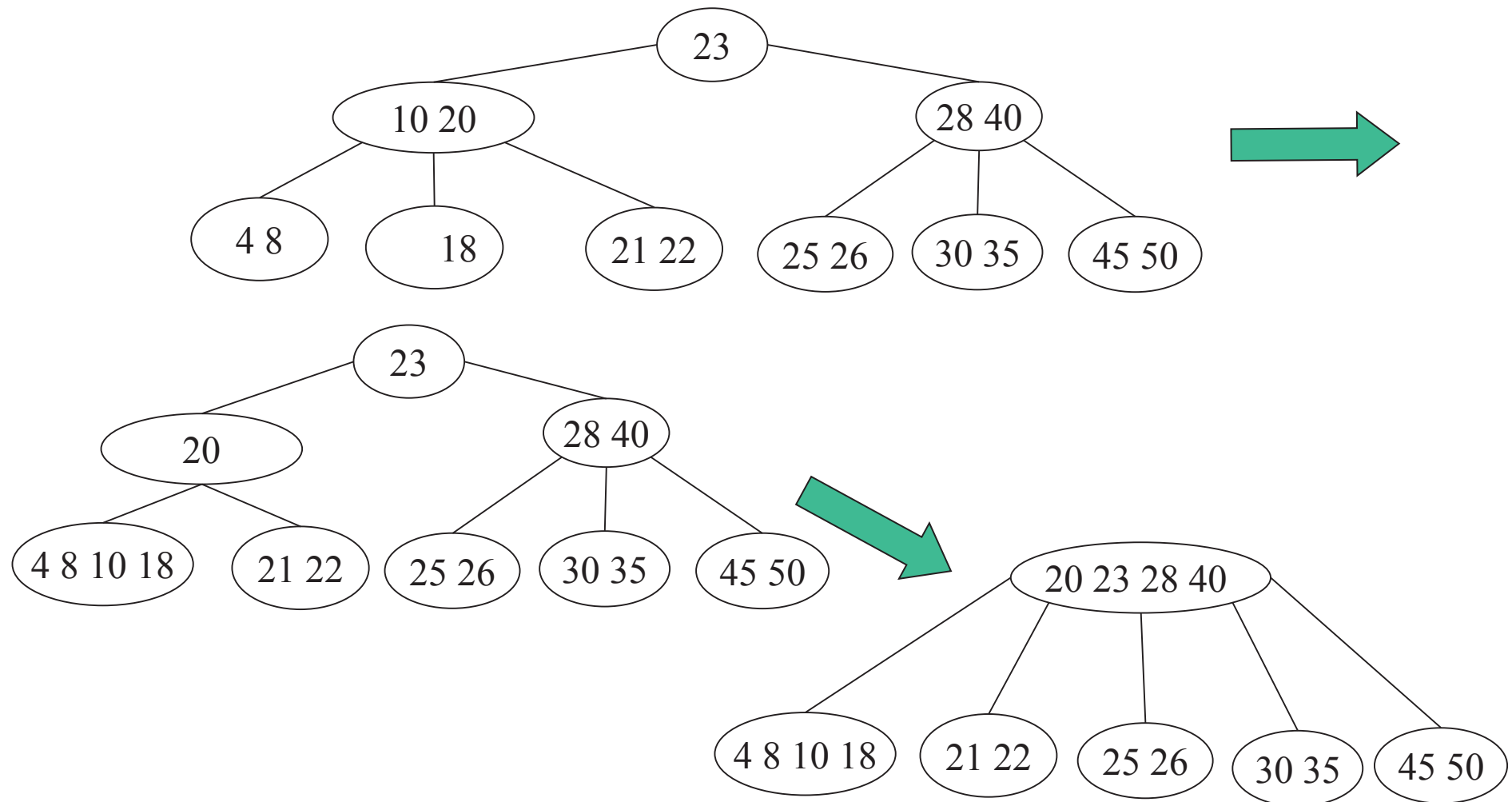
Ordem 2

Registro 13



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

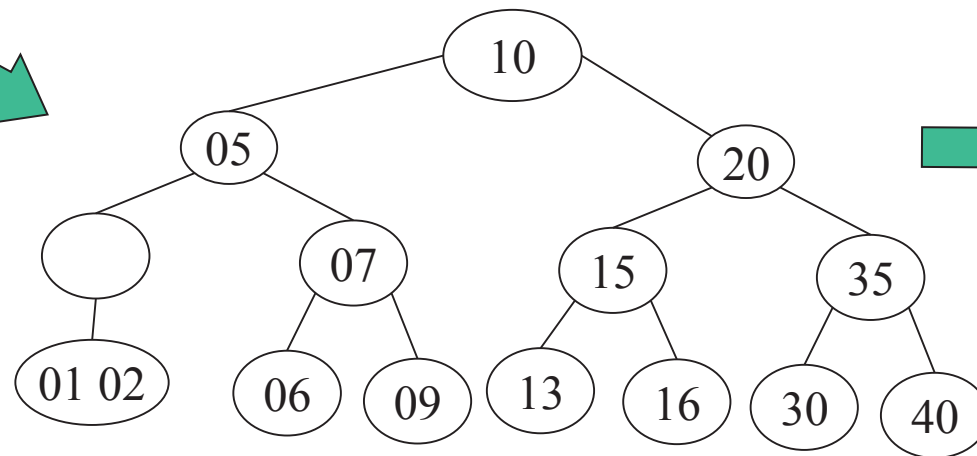
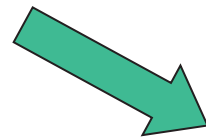
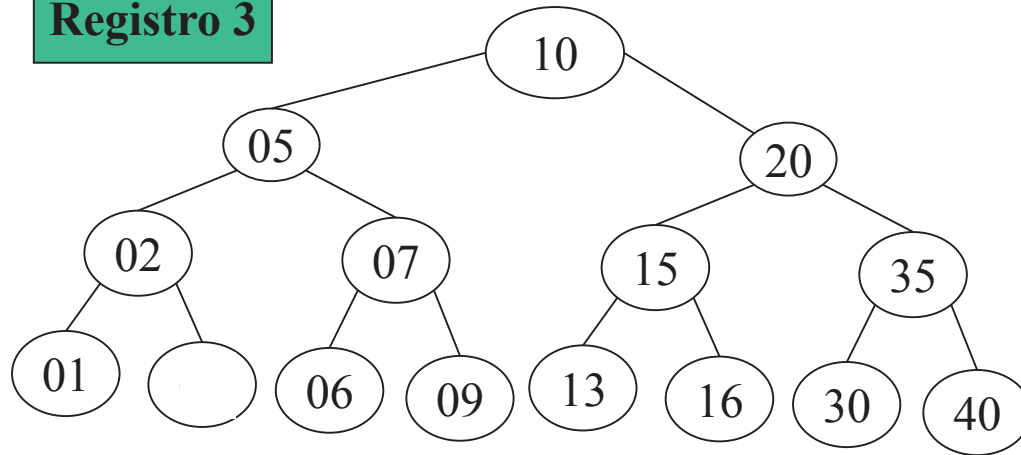
Remoção de Registros



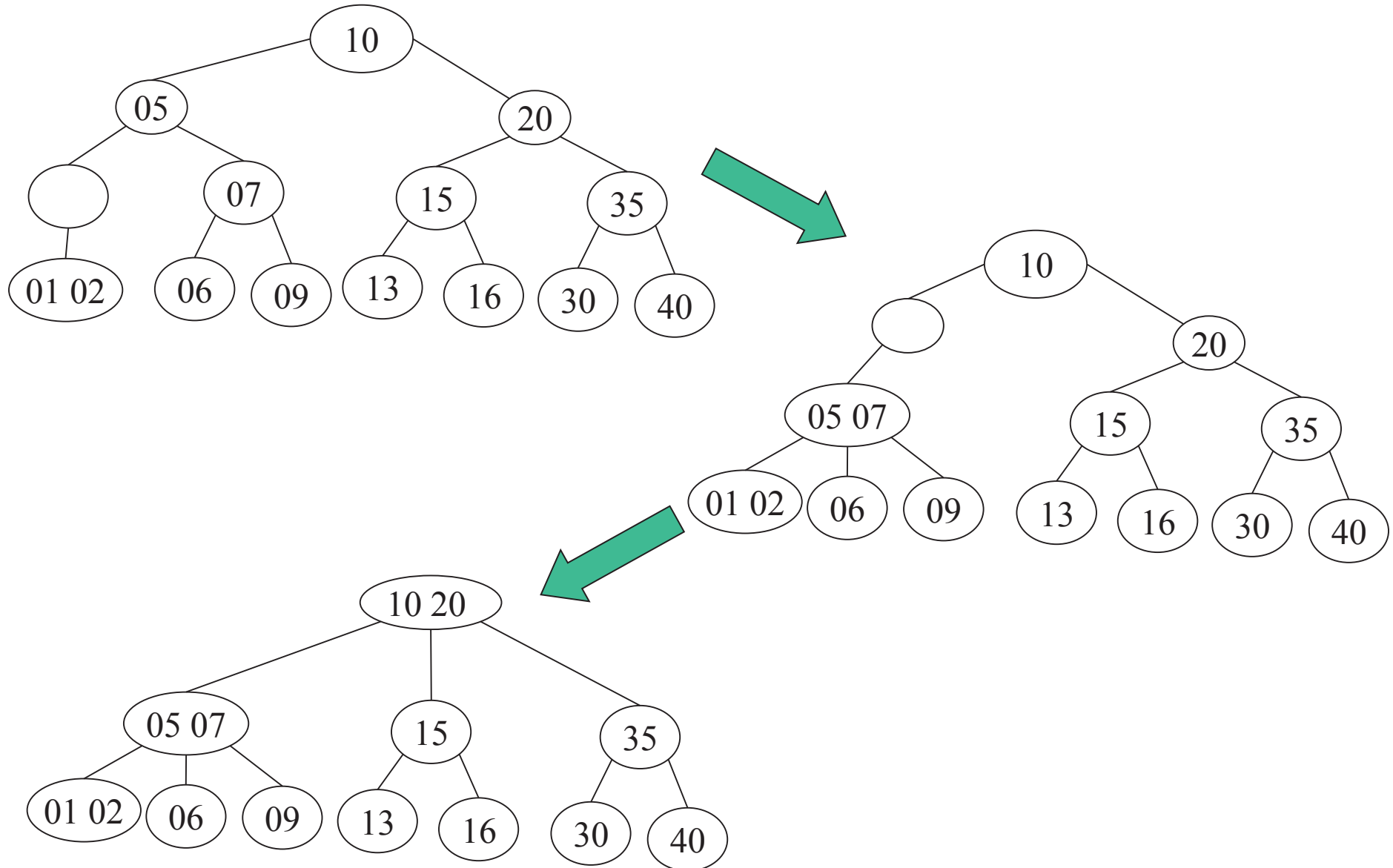
Remoção de Registros

Outro exemplo: Árvore B de ordem 1

Registro 3



Remoção de Registros



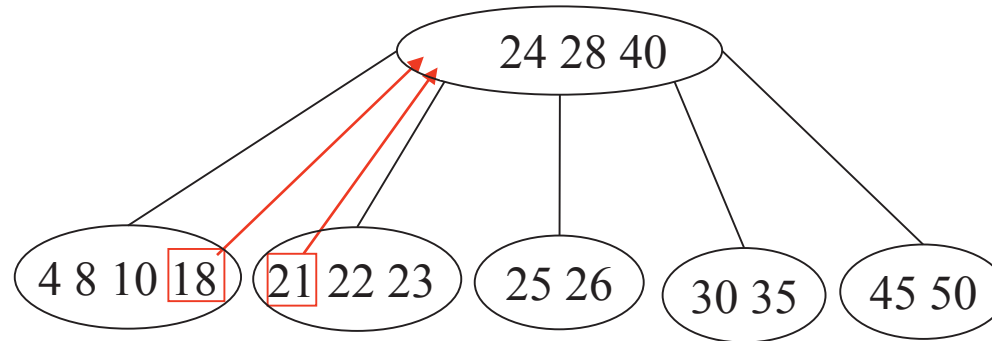
Remoção

- **Caso 2: quando o registro não se encontra em uma folha.**
- **Mesmo procedimento de remoção em árvores binárias ou AVL**
 - Substituição pelo registro mais à direita da sub-árvore à esquerda;
 - Substituição pelo registro mais à esquerda 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**

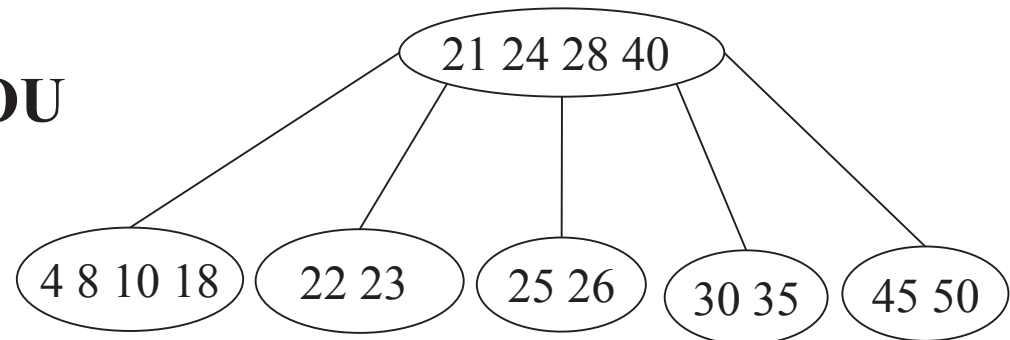
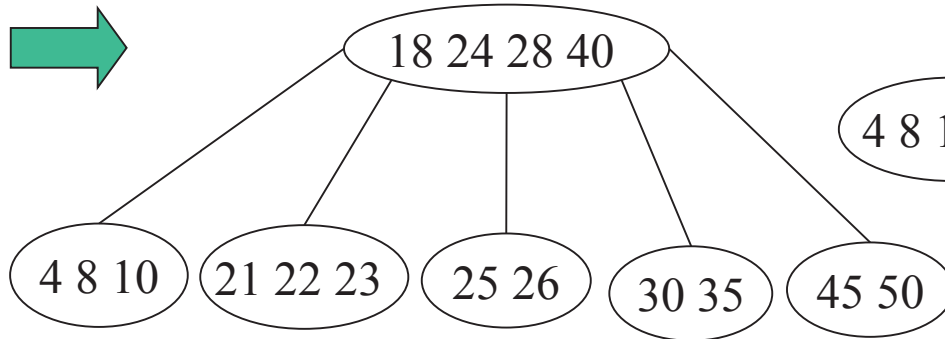
Remoção de Registros

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

Registro 20



OU

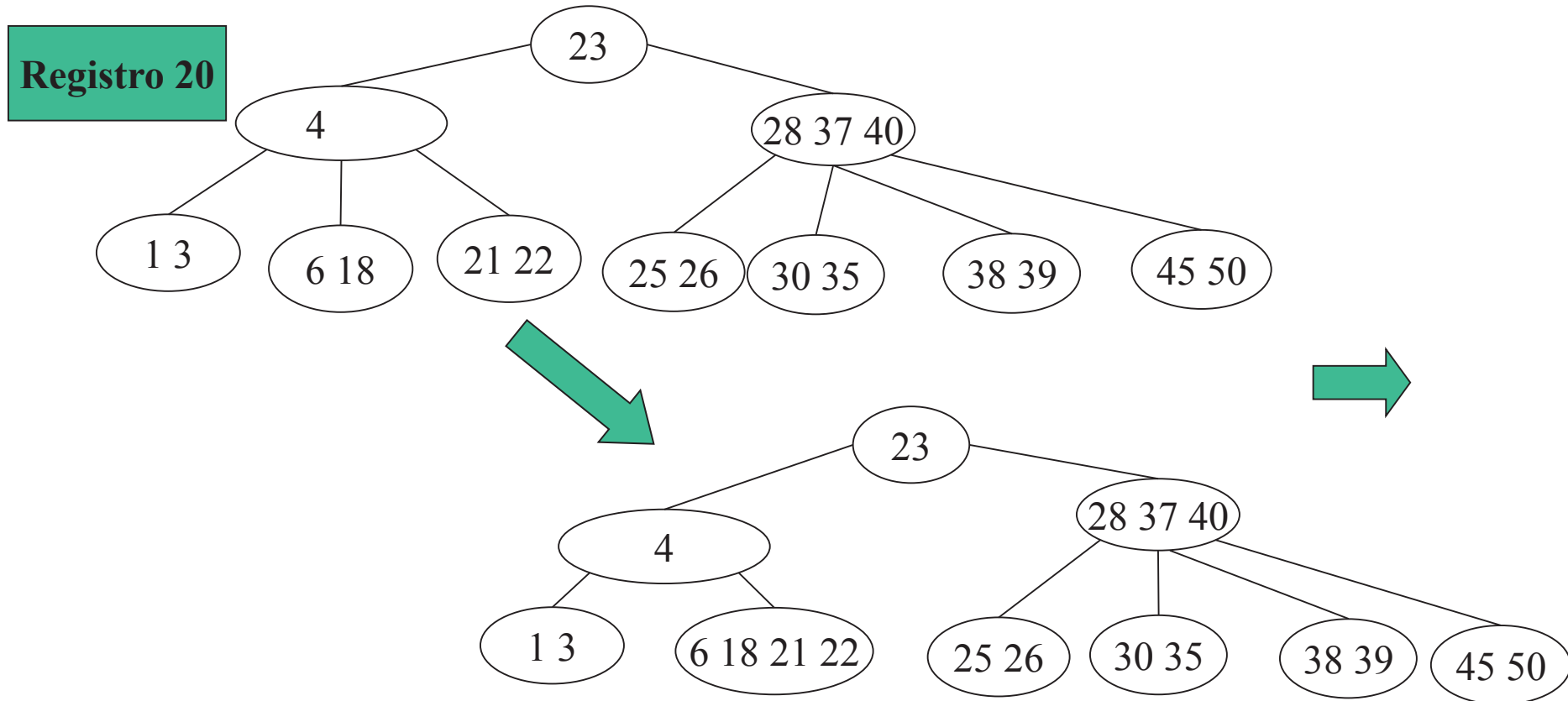


Remoção de Registros

- **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**

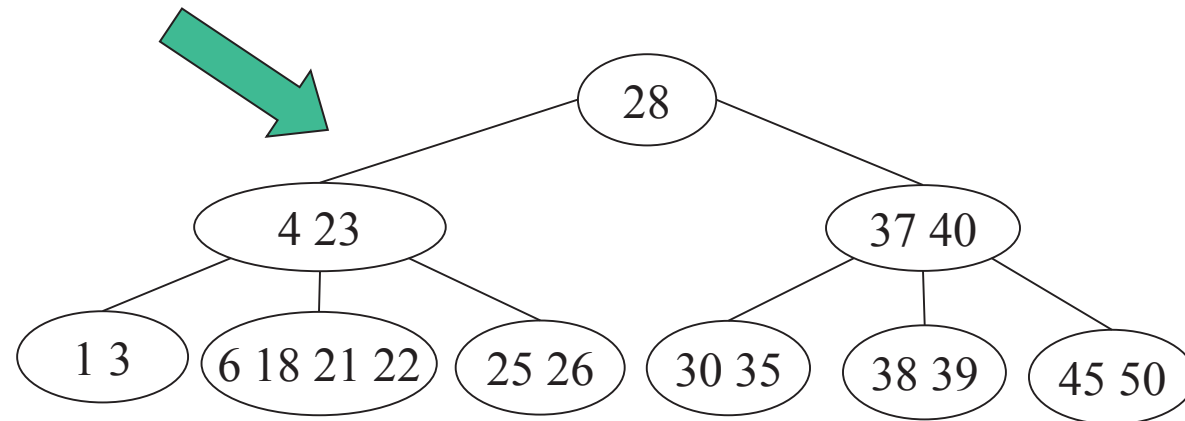
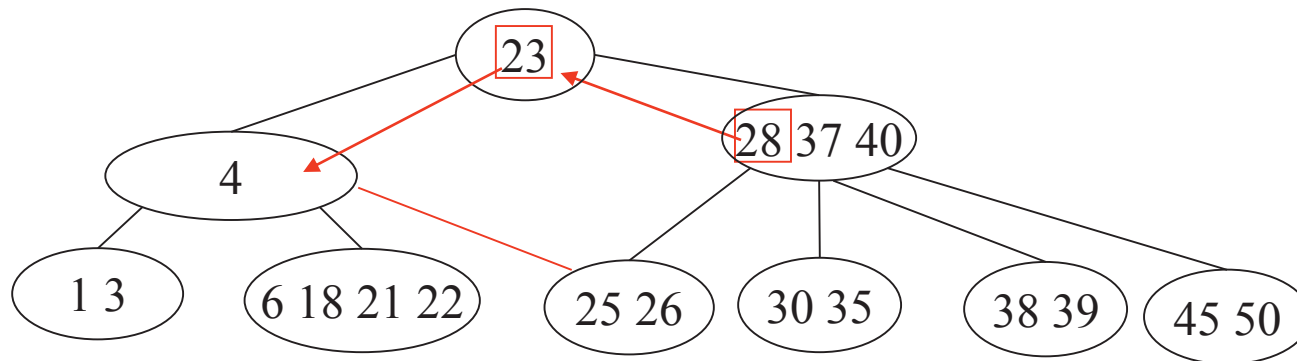
Remoção de Registros

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



Concatena-se o filho à esquerda e à direita do registro excluído; o registro mais à esquerda (ou mais à direita) do irmão é promovido para a raiz e o registro à esquerda (ou à direita) da raiz é inserido na página que teve o registro excluído; o filho à esquerda (ou o filho à direita) do registro que foi emprestado passa a ser o filho à direita (ou à esquerda) do nó para onde ele foi deslocado;

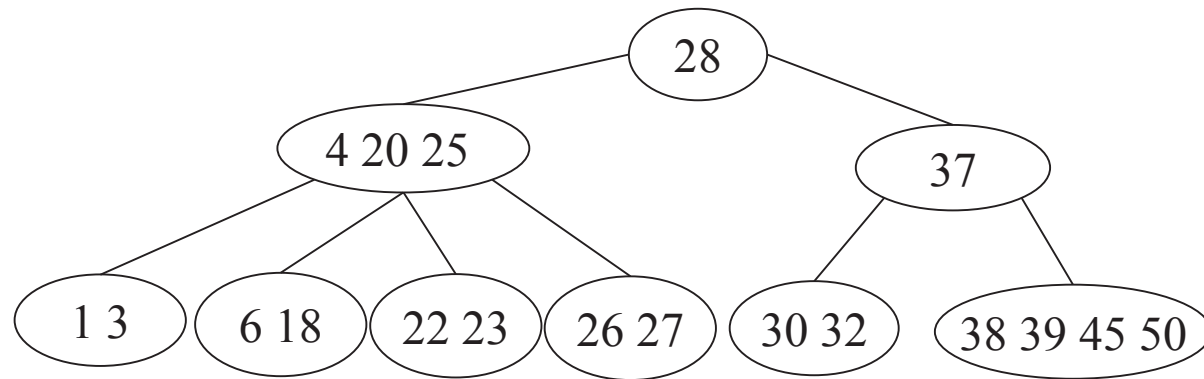
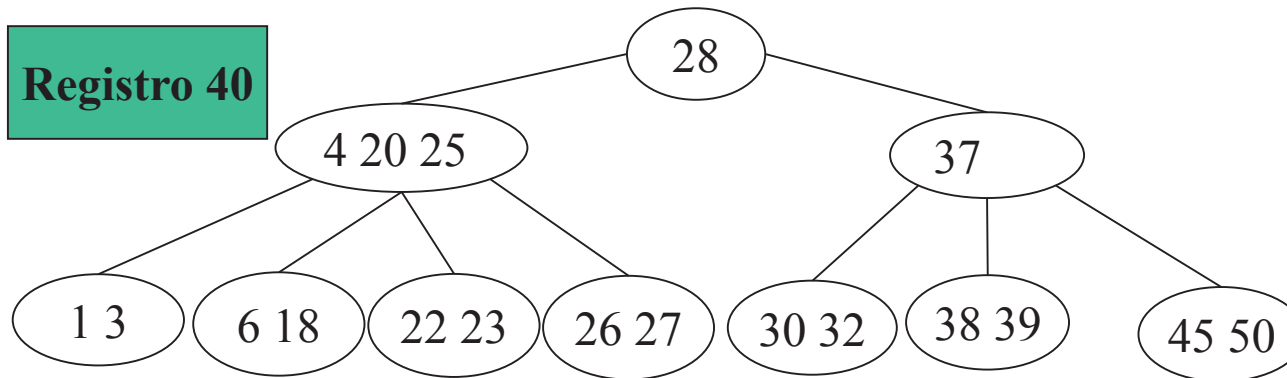
Remoção de Registros



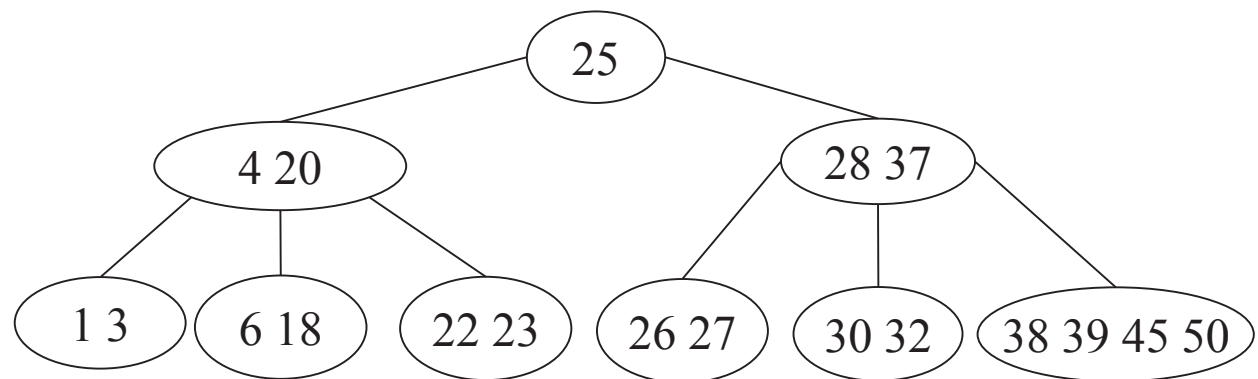
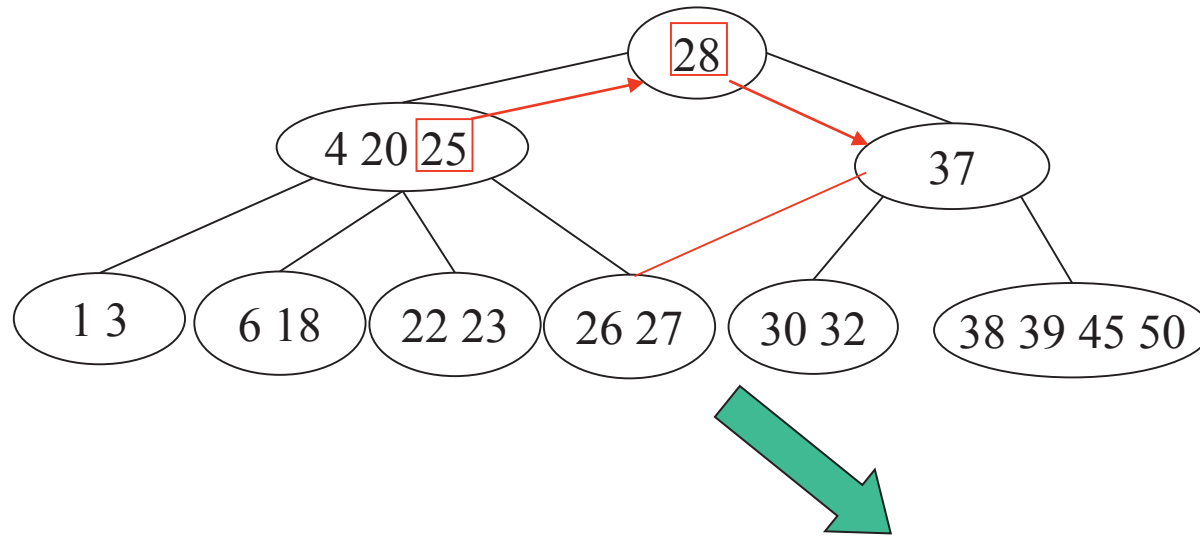
Remoção de Registros

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

Registro 40



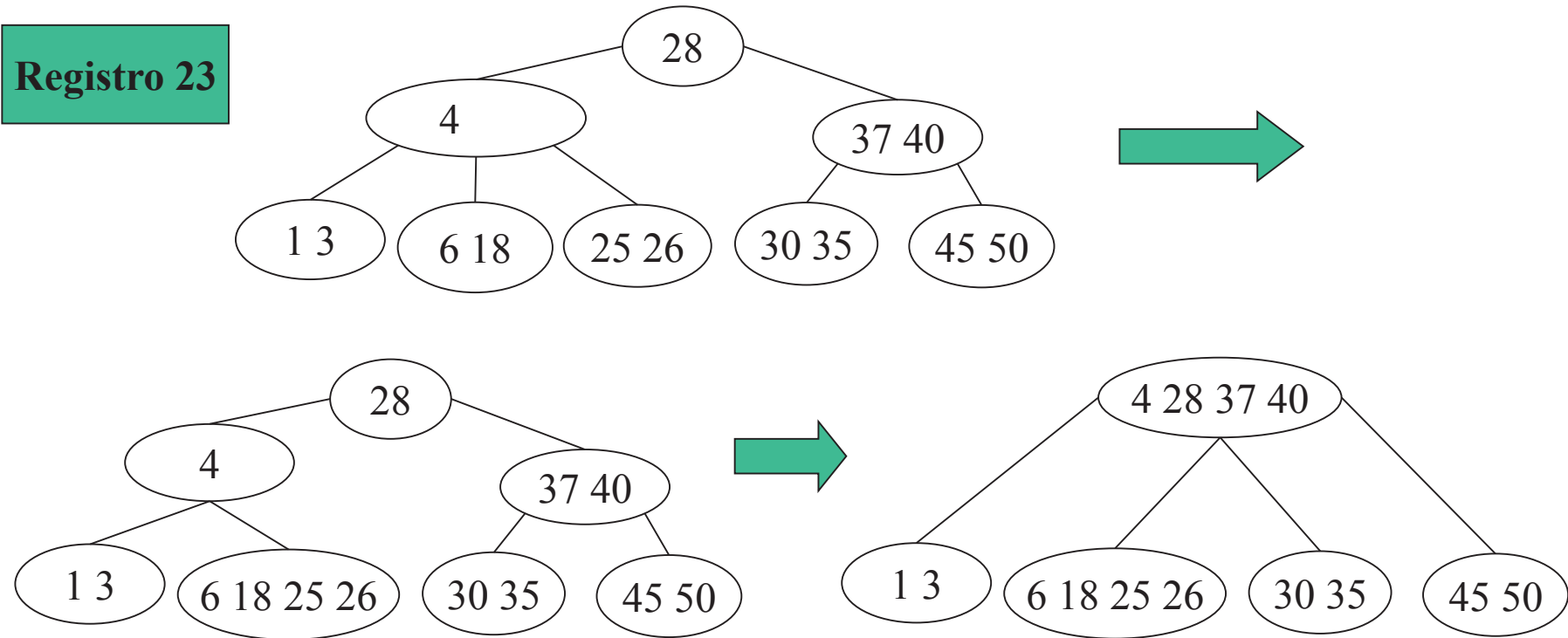
Remoção de Registros



Remoção de Registros

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

Registro 23



Concatena-se os registros filhos à esquerda e à direita do registro excluído; em seguida, concatena-se os registros da página excluída, do seu irmão e do seu pai

Árvores B

- Considerações finais:
 - Algoritmos de pesquisas são bem mais simples;
 - Algoritmos de inserção e exclusão podem ser complicados, dependendo da ocasião;
 - Não se aprende árvores B sem praticar;
 - Sugestão: <http://slady.net/java/bt/view.php>;

Árvores B

- Praticando:
 - Faça em forma de desenho o passo a passo para efetuar a inserção dos seguintes elementos em uma árvore B: 20,26,12,5,33,17,45,2,10,52,40;

Algoritmos e Estruturas de Dados III

- Bibliografia:

- Básica:

- ASCENCIO, Ana C. G. Estrutura de dados. Rio de Janeiro: Pearson. 2011.
 - CORMEN, Thomas; RIVEST, Ronald; STEIN, Clifford; LEISERSON, Charles. Algoritmos. Rio de Janeiro: Elsevier, 2002.
 - ZIVIANI, Nívio. Projeto de algoritmos com implementação em Pascal e C. São Paulo: Cengage Learning, 2010.

- Complementar:

- EDELWEISS, Nina, GALANTE, Renata. Estruturas de dados. Porto Alegre: Bookman. 2009. (Coleção Livros didáticos de informática UFRGS, 18).
 - PINTO, W.S. Introdução ao desenvolvimento de algoritmos e estrutura de dados. São Paulo: Érica, 1990.
 - PREISS, Bruno. Estruturas de dados e algoritmos. Rio de Janeiro: Campus, 2000.
 - TENEMBAUM. Aaron M. Estruturas de Dados usando C. São Paulo: Makron Books. 1995.
 - VELOSO, Paulo A. S. Complexidade de algoritmos: análise, projeto e métodos. Porto Alegre: Sagra Luzzatto, 2001.

Algoritmos e Estruturas de Dados III

