



CENTRO DE CIÊNCIA E TECNOLOGIA
LABORATÓRIO DE CIÊNCIAS MATEMÁTICAS
UNIVERSIDADE ESTADUAL DO NORTE FLUMINENSE

Árvores 2-3-4 Árvores 2-3

Disciplina: Estrutura de Dados II

Prof. Fermín Alfredo Tang Montané

Curso: Ciência da Computação

Árvores B

Definição

- Uma árvore B de ordem m é uma árvore de busca múltipla (ou multicaminho), onde m representa o número máximo de filhos de um nó da árvore, em que se atendem as seguintes condições:
 1. O nó raiz possui l entradas (chaves), onde $1 \leq l \leq m - 1$. Esse nó poderá ser uma folha ou ter k filhos (subárvores), onde $2 \leq k \leq m$.
 2. Cada nó interno possui k filhos, onde $\lceil m/2 \rceil \leq k \leq m$; e contêm l entradas, onde $\lceil m/2 \rceil - 1 \leq l \leq m - 1$.
 3. Cada nó folha não possui filhos; e contêm l entradas, sendo que $\lceil m/2 \rceil - 1 \leq l \leq m - 1$.
 4. Todas as folhas estão no mesmo nível.
 5. Em cada nó as chaves estão ordenadas.

Árvores B

Casos Particulares

- Embora as árvore B tenham sido concebidas para armazenar uma grande quantidade itens de dados em cada nó, existem casos especiais com um número pequeno de elementos por nó.
- Este tipo de árvores foram as primeiras tentativas de desenvolver árvores balanceadas sem o conceito de rotação das árvores AVL.
- Posteriormente, mostraremos que existe equivalência entre um tipo particular de árvore B e as árvores binárias balanceadas.

Árvores 2-3-4

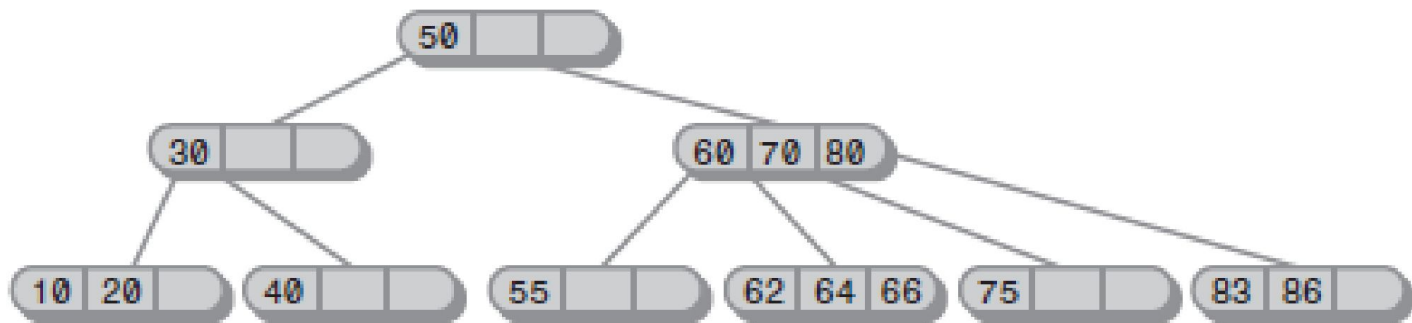
Definição

- Uma árvore 2-3-4 é um caso especial de árvore B de ordem 4.
- As árvores 2-3-4, recebem o seu nome em referencia ao número de filhos que cada nó pode ter. Qualquer nó que não seja uma folha, pode apresentar três configurações:
 - um nó com apenas uma entrada (ou chave) sempre tem dois filhos;
 - um nó com duas entradas (ou chaves) sempre tem três filhos;
 - um nó com três entradas (ou chaves) sempre tem quatro filhos.
- Essa árvore foi discutida inicialmente por Rudolf Bayer, que a chamou de árvore B binária simétrica (1972) mas ela é usualmente chamada de árvore 2-3-4 ou apenas árvore 2-4.

Árvores 2-3-4

Exemplo

- A figura mostra um exemplo de árvore 2-3-4. Esta árvore é uma árvore B de ordem $m = 4$.



- Observe que trata-se de uma árvore B de pequeno porte.
 - a raiz possui k filhos, onde $2 \leq k \leq 4$ e l entradas, onde $1 \leq l \leq 3$;
 - cada nó interno possui k filhos, onde $2 \leq k \leq 4$ e l entradas, $1 \leq l \leq 3$;
 - cada nó folha contém l entradas, onde, $1 \leq l \leq 3$;
 - todas as folhas estão no mesmo nível.
 - as chaves estão ordenadas em cada nó.

Árvores 2-3-4

Exemplo

- Uma árvore 2-3-4 é um caso particular de árvore B de ordem 4.

Variação do número
de filhos:

$$\lceil m/2 \rceil \leq k \leq m$$

$$\lceil 4/2 \rceil \leq k \leq 4$$

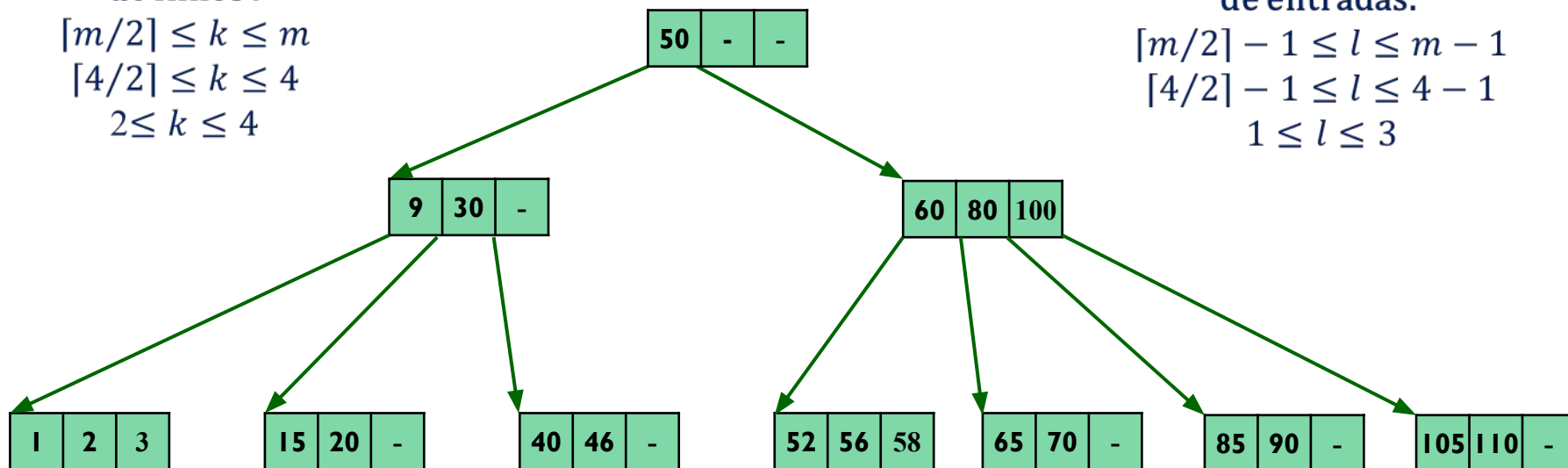
$$2 \leq k \leq 4$$

Variação do número
de entradas:

$$\lceil m/2 \rceil - 1 \leq l \leq m - 1$$

$$\lceil 4/2 \rceil - 1 \leq l \leq 4 - 1$$

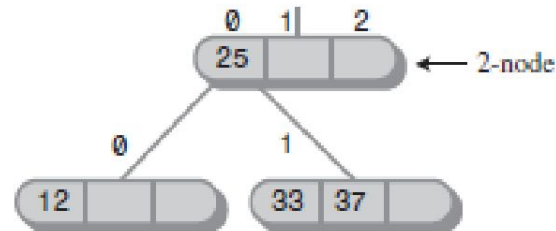
$$1 \leq l \leq 3$$



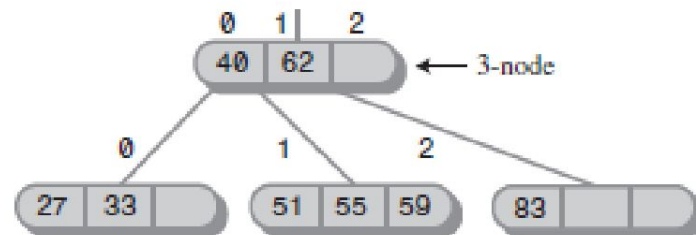
Árvores 2-3-4

Exemplo

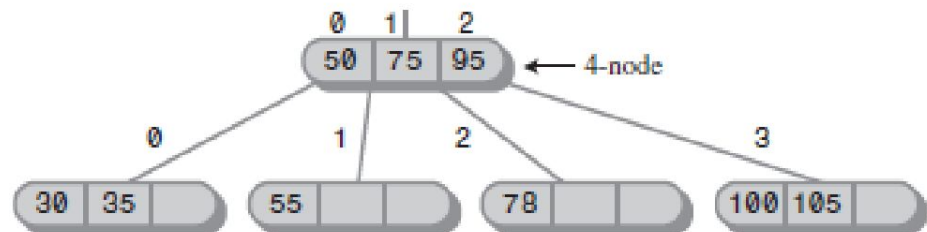
- A figura mostra as possibilidades. Por conveniência numeramos os elementos de 0 a 2 e os nós filhos de 0 a 3.



- Um nó com duas ligações é chamado de nó-2.
- Um nó com três ligações é um nó-3;
- Um nó com quatro ligações é um nó-4.



- Observe que neste tipo de árvore não existe algo como um nó-1.



Nodes in a 2-3-4 tree.

Árvores 2-3-4

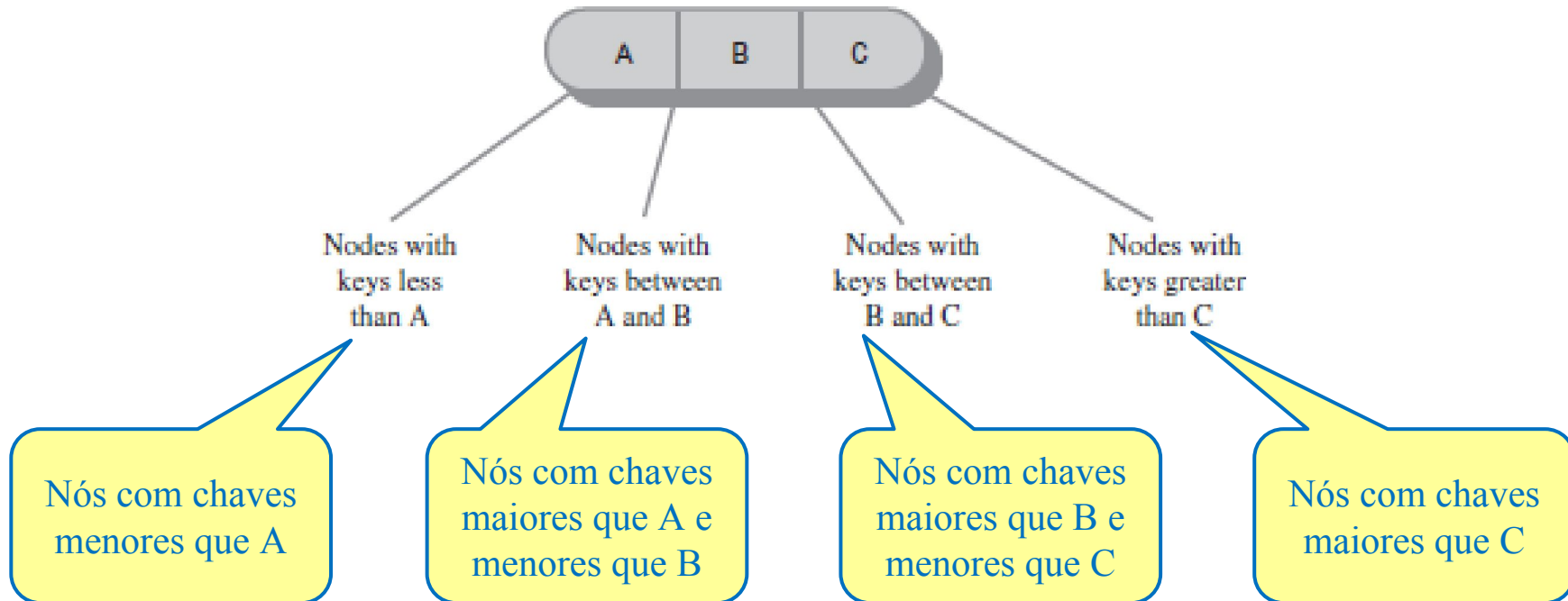
Organização

- Um aspecto importante da estrutura de qualquer árvore é a relação de seus nós com os valores-chave de seus elementos.
- Em uma árvore 2-3-4, segue-se o mesmo princípio de uma árvore binária, porém generalizado:
 - Todos os filhos na subárvore com raiz no filho 0 tem valores-chave menores que a chave 0.
 - Todos os filhos na subárvore com raiz no filho 1 tem valores-chaves maiores que a chave 0, porém menores que a chave 1.
 - Todos os filhos na subárvore com raiz no filho 2 tem valores-chaves maiores que a chave 1, porém menores que a chave 2.
 - Todos os filhos na subárvore com raiz no filho 3 tem valores-chaves maiores que a chave 2.
- Valores duplicados geralmente não são permitidos nas árvores 2-3-4, portanto, não precisamos nos preocupar em comparar chaves iguais.

Árvores 2-3-4

Organização

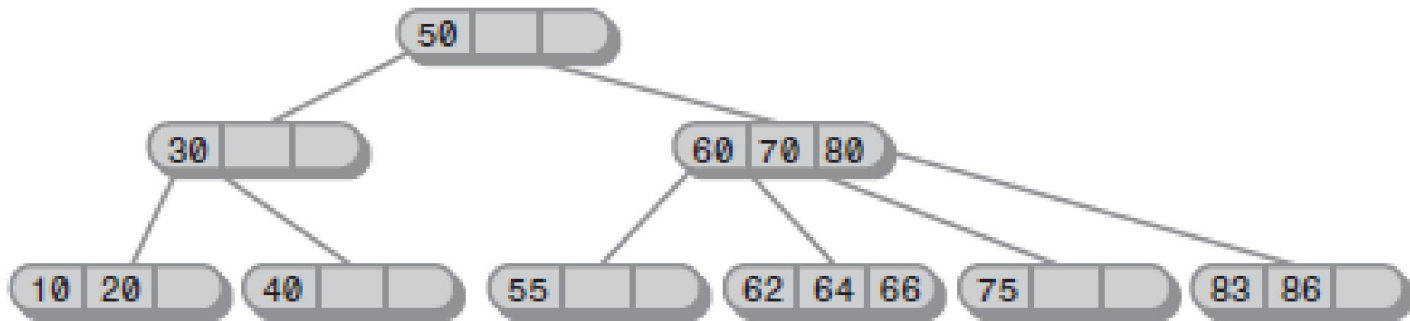
- A figura ilustra a relação entre os nós com base nos valores-chaves de seus itens de dados.



Árvores 2-3-4

Organização

- No exemplo, podemos ver a relação entre os nós sendo atendida.



- Nesta árvore observa-se que:
 - Todas as folhas estão no mesmo nível. O nível inferior, que é o nível 2.
 - Os nós não estão todos completos. Alguns nós possuem apenas um ou dois elementos, ao invés de três.
 - A árvore está balanceada e se manterá balanceada independente da sequência de dados inserida ou removida.

Árvores 2-3-4

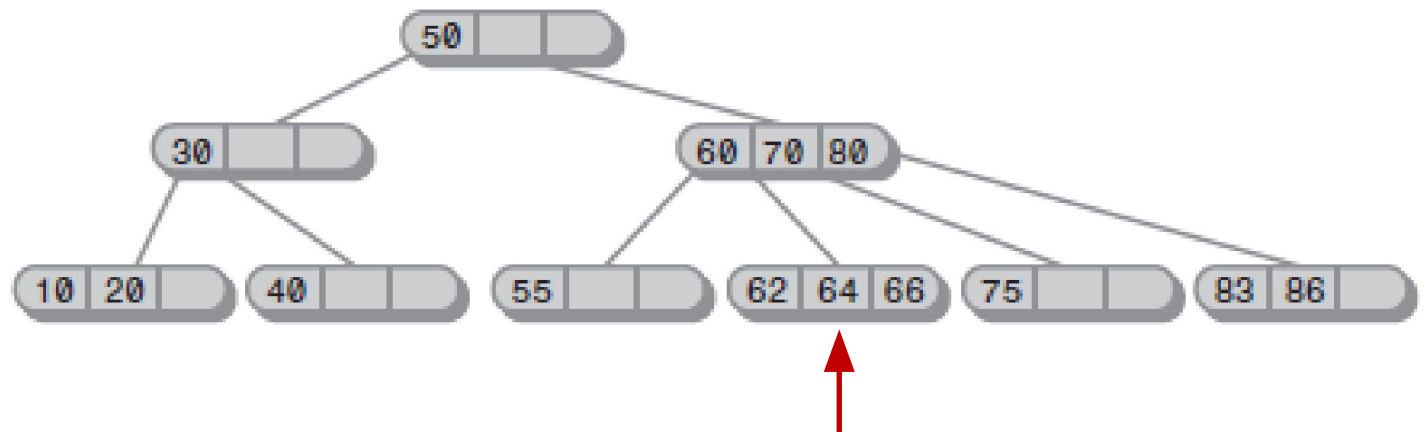
Busca

- Localizar um item de dados com uma determinada chave é similar ao processo de busca em uma árvore binária, só que de forma generalizada.
- Começa-se pelo nó raiz e a menos que a chave de busca seja encontrada nesse nó, seleciona-se a ligação que leva à subárvore que contém nós com chaves dentro da faixa de valores que compreenda a chave procurada.

Árvores 2-3-4

Busca

- Para buscar o item de dados com **chave 64** na árvore de exemplo, começa-se pela raiz. Busca-se na raiz, mas o item não é encontrado lá.
- Como 64 é maior que 50, iremos ao **filho I**, representado como 60/70/80. O item de dados também não se encontra neste nó, portanto terá que ir para o próximo filho.
- Como 64 é maior que 60, porém menor que 70, irá novamente para o **filho I**. Desta vez, encontrará o item especificado no nó 62/64/66.



Árvores 2-3-4

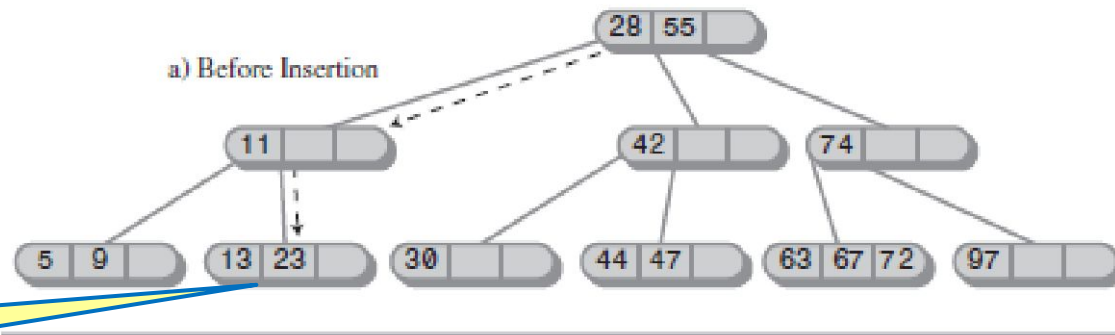
Inserção

- Novos itens de dados são sempre inseridos em folhas, que estão no nível inferior da árvore.
- A inserção em uma árvore 2-3-4 é às vezes muito simples e outras vezes mais complexo. Em ambos os casos, o processo começa por buscar o nó folha destino, que contém a faixa de valores do nó inserido.
- **Inserção sem divisão.-**
- O caso mais simples é chamado de inserção sem divisão.
- Se nenhum nó completo (ou cheio) for encontrado durante a busca, a inserção será simples. Quando o nó folha destino for atingido, o elemento simplesmente será inserido nele.
- A inserção pode envolver deslocar um ou dois elementos em um nó para que as chaves estejam na ordem correta após o novo elemento ser inserido.

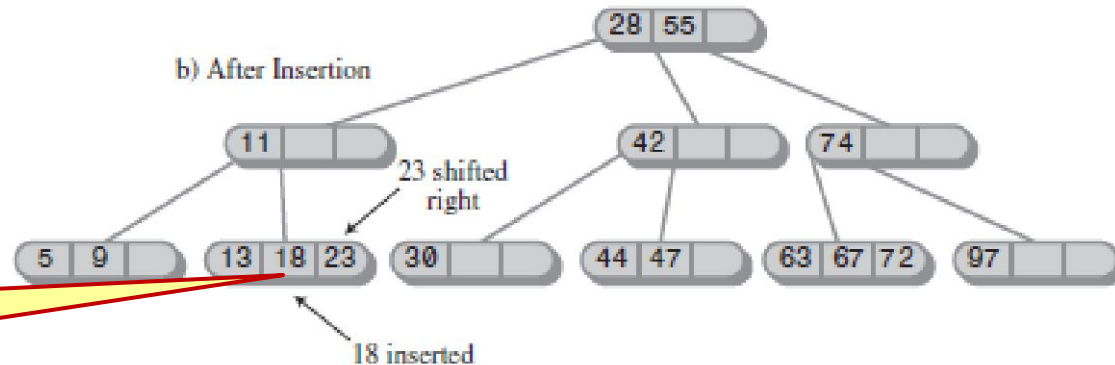
Árvores 2-3-4

Inserção sem divisão

- A figura ilustra o caso simples de inserção. Um elemento com chave 18 é inserido na árvore 2-3-4 da figura a).



Nó destino



Deslocamento e
inserção

- O novo item é inserido no nó 13/23. Para isso, o item 23 teve que ser deslocado para a direita para criar espaço para o item 18.

Árvores 2-3-4

Inserção com divisão

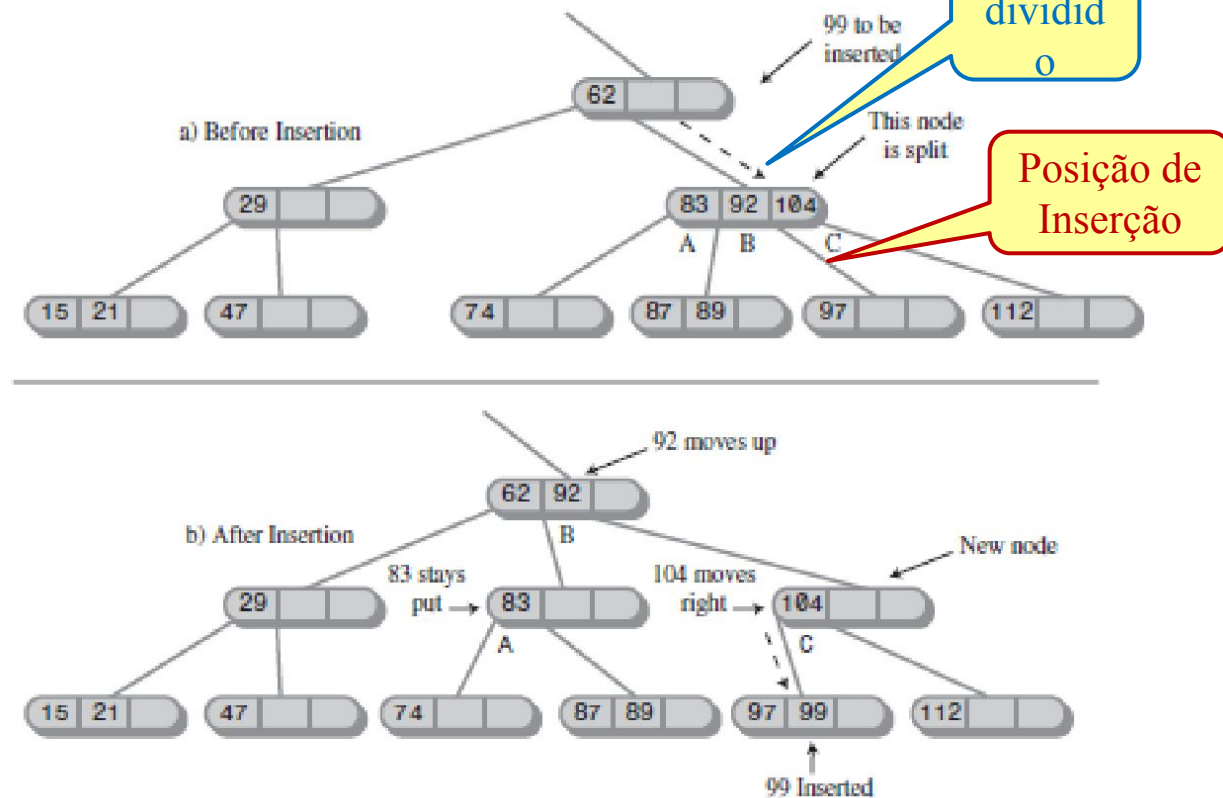
- **Inserção com divisão.-**
- O caso mais complexo é chamado de inserção com divisão.
- Este caso acontece se um nó completo (cheio) for encontrado no caminho até o ponto de inserção. Quando isto acontece o nó tem que ser dividido.
- É este processo de divisão que mantém a árvore balanceada.
- O tipo de árvore 2-3-4 que estamos descrevendo é chamado de **árvore 2-3-4 de cima para baixo**, porque os nós são divididos ao descer até o ponto de inserção.
- Vamos chamar os elementos do nó que será dividido de A, B e C. Assumindo que o nó dividido não seja uma raiz. Temos que:
 - Um novo nó vazio é criado. Ele é um irmão do nó sendo dividido e é colocado a sua direita.
 - O item de dados C é movido para o novo nó.
 - O item de dados B é movido para o nó pai do nó sendo dividido.
 - O item de dados A permanece onde está.
 - Os dois filhos mais à direita são desconectados do nó dividido e conectados ao novo nó.

Árvores 2-3-4

Inserção com divisão

- A figura ilustra o caso de inserção com divisão. Um elemento com chave 99 é inserido na árvore 2-3-4 da figura a) resultando na árvore da figura b).

- Observe que o efeito da divisão de um nó é mover dados para cima e para a direita. Essa reorganização que mantém a árvore balanceada.
- Aqui a inserção exigiu apenas uma divisão de nó, porém mais de um nó completo pode ser encontrado no caminho até o ponto de inserção.
- Quando for o caso, haverá diversas divisões.



- Observe que um nó-4 foi transformado em dois nós-2.

Árvores 2-3-4

Divisão da Raiz

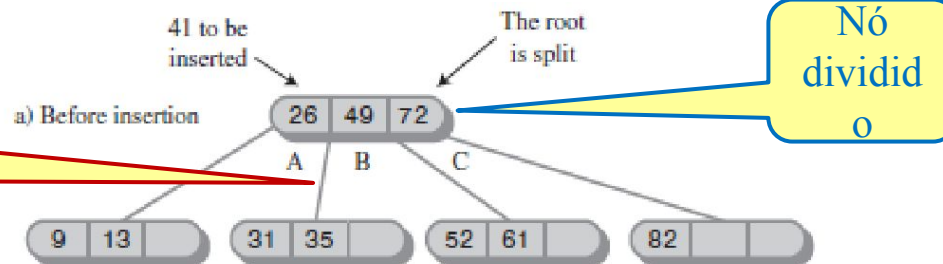
- **Divisão da raiz.-**
- Quando uma **raiz cheia** for encontrada no início da busca pelo ponto de inserção, a inserção resultante será um pouco mais complexa. Neste caso:
 - Uma nova raiz é criada. Ele torna-se pai do nó sendo dividido.
 - Um segundo nó é criado. Ele torna-se irmão do nó sendo dividido e colocado à direita.
 - O item de dados C é movido para o novo nó irmão.
 - O item de dados B é movido para a nova raiz.
 - O item de dados A permanece onde está.
 - Os dois filhos mais a direita são desconectados do nó dividido e conectados ao novo nó à direita.

Árvores 2-3-4

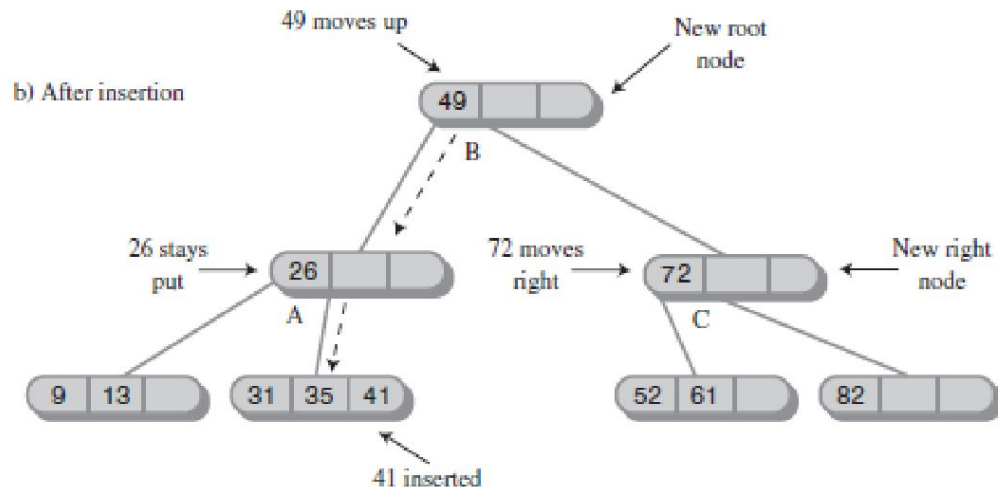
Divisão da raiz

- A figura ilustra o caso de inserção com divisão da raiz. Um elemento com chave 41 é inserido na árvore 2-3-4 da figura a) resultando na árvore da figura b).

Posição de Inserção



- Este processo cria uma nova raiz que está em um nível mais alto que a antiga.
- A altura da árvore é aumentada em um.
- A busca pelo ponto de inserção continua ao descer na árvore.



- Observe que um nó-4 foi transformado em três nós-2.

Árvores 2-3-4

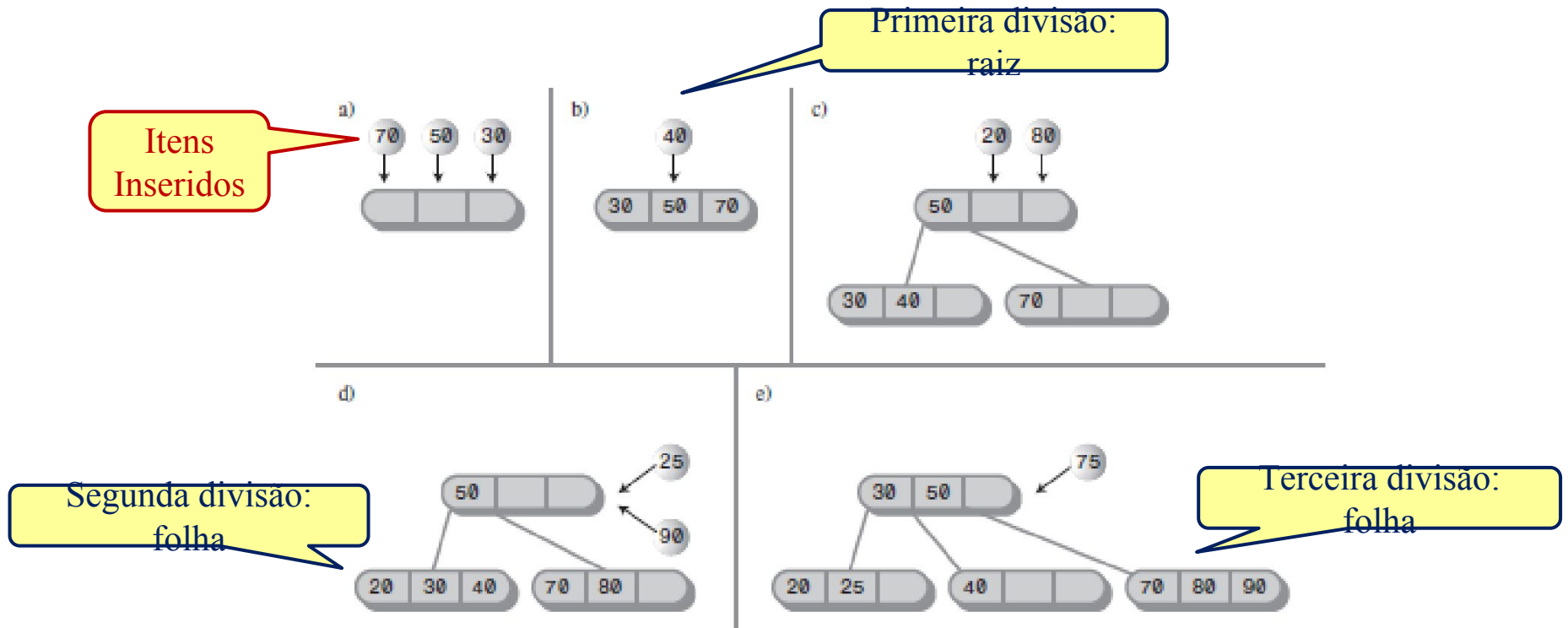
Dividindo ao descer

- Observe que, como os nós completos são divididos ao descer, neste caso, uma divisão não pode se propagar para cima na árvore.
- O pai de qualquer nó que esteja sendo dividido tem a garantia de não estar completo. Se estivesse completo teria sido dividido na primeira visita. Portanto, pode aceitar o elemento B sem precisar ser dividido.
- Vale observar que se esse pai já tinha dois filhos quando seu filho foi dividido, então ele se tornará completo. Porém, isso significa apenas que ele será dividido.

Árvores 2-3-4

Exemplos

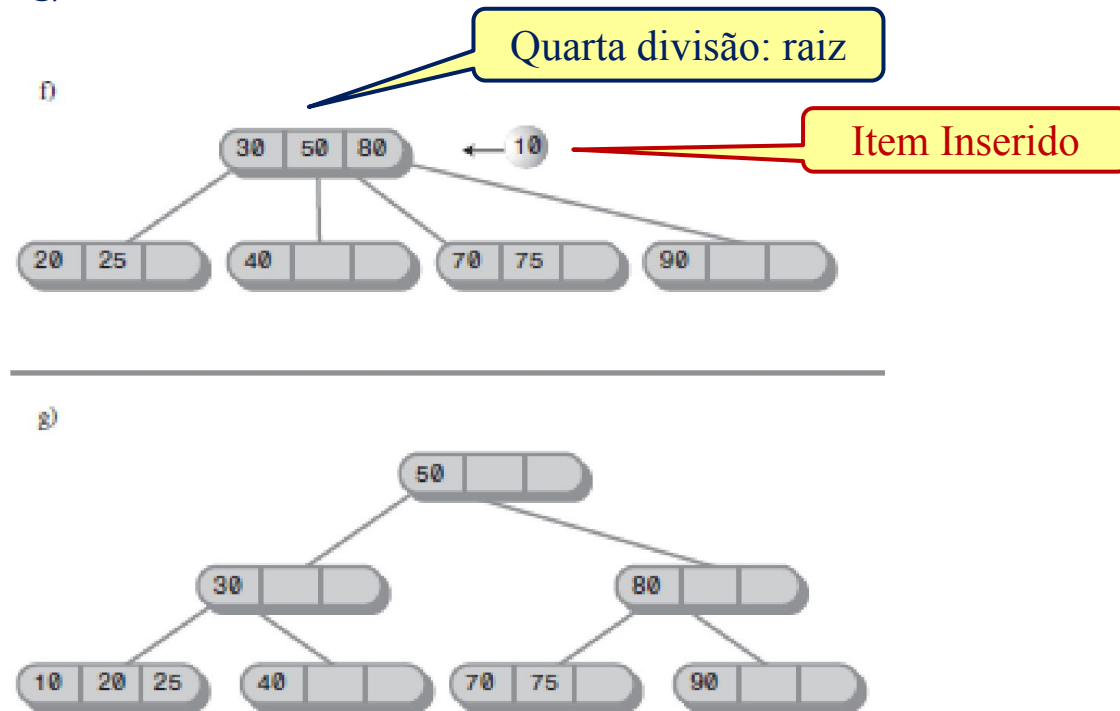
- A figura ilustra uma série de inserções em uma árvore vazia.
- Há quatro divisões de nós. Duas da raiz e duas das folhas.



Árvores 2-3-4

Exemplos

- As figuras ilustram a última inserção. A inserção do item 10 na árvore da figura f). O nó raiz é dividido, o item 10 inserido em uma folha, resultando na árvore da figura g).



Árvores Vermelho-Preta

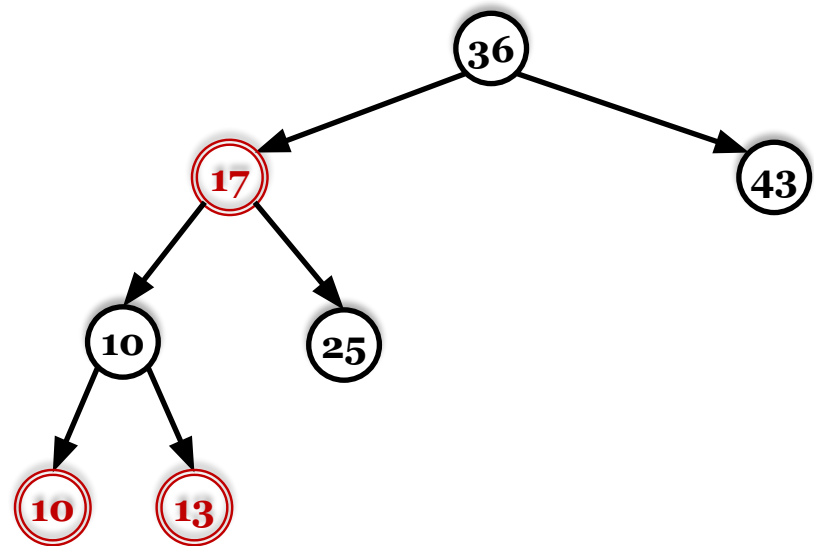
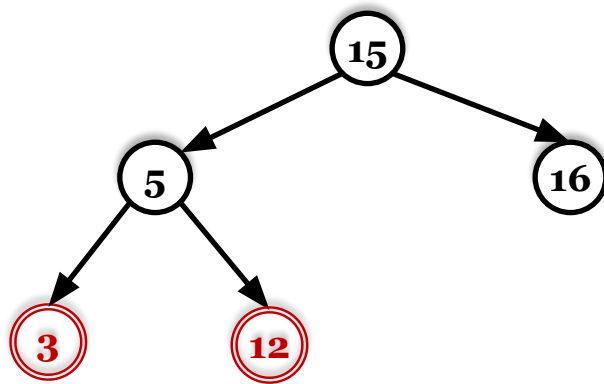
Definição

- Uma árvore vermelho-preta (rubro-negra) é um tipo especial de árvore binária que possui um certo grau de balanceamento.
- O balanceamento é conseguido durante a inserção ou remoção de elementos verificando-se certas características. Caso essas características sejam violadas uma ação corretiva será tomada, estruturando de novo a árvore.
- **Regras vermelho-pretas:**
 1. Cada nó é vermelho ou preto.
 2. O nó raiz é sempre preto.
 3. Se um nó for vermelho, seus filhos terão que ser pretos (o contrário não é exigido).
 4. Todo caminho da raiz para uma folha (ou para um filho nulo) tem que conter o mesmo número de nós pretos.
- O número de nós pretos em um caminho da raiz para uma folha é chamado de altura preta. A regra 4 exige que a altura preta seja a mesma para todos os caminhos da raiz até uma folha. Estas regras parecem muito misteriosas mas farão mais sentido depois

Árvores Vermelho-Preta

Exemplo

- As figuras ilustram exemplos de árvores vermelho-pretas.



Árvores 2-3-4 vs Vermelho-Preta

Relação

- Enquanto as árvores 2-3-4 são árvores multi-caminhos as árvores vermelho-preta (rubro negras) são árvores binárias balanceadas.
- Existe alguma relação entre elas?
- De fato existe uma relação importante.
- Um tipo de árvore pode ser transformada na outra pela aplicação de algumas regras simples. Até as operações necessárias para mantê-las balanceadas são equivalentes. Matemáticos diriam que elas são isomórficas.
- Historicamente, a árvore 2-3-4 foi desenvolvida primeiro; mais tarde a árvore vermelho-preta desenvolveu-se a partir dela.

Árvores 2-3-4 e Vermelho-Preta

Transformação

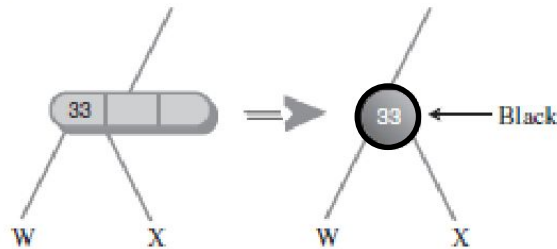
- Uma árvore 2-3-4 pode ser transformada em uma árvore vermelho-preta aplicando-se três regras:
 - Transforme qualquer nó-2 na árvore 2-3-4 em um nó preto na árvore vermelho-preta;
 - Transforme qualquer nó-3 na árvore 2-3-4 em dois nós na árvore vermelho-preta. Um nó filho e um nó pai. Não importa qual item se torna-se filho e qual torna-se pai. Assim, existem duas formas de se fazer isso. O filho é colorido de vermelho e o pai colorido de preto.
 - Transforme qualquer nó-4 na árvore 2-3-4 em três nós na árvore vermelho-preta. Dois nós filhos e um nó pai. Os filhos são coloridos de vermelho e o pai colorido de preto.
- Uma forma de entender a coloração é a seguinte: A coloração preta indica que esse nó é independente, embora possa ter filhos vermelhos que dependem dele. A coloração vermelha indica que esse nó depende de um pai preto.

Árvores 2-3-4 e Vermelho-Preta

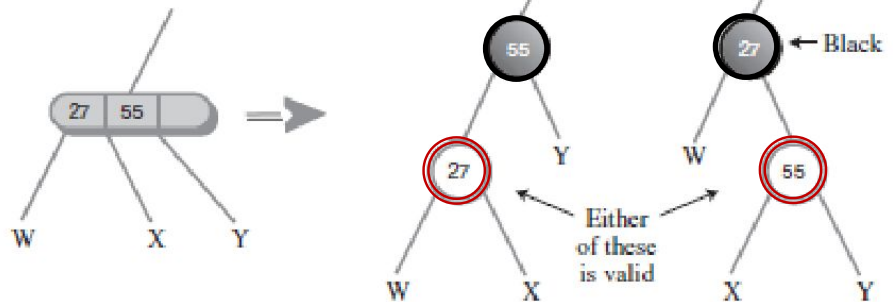
Transformação

- As três regras de transformação de uma árvore 2-3-4 em uma árvore vermelho-preta são ilustradas na figura:

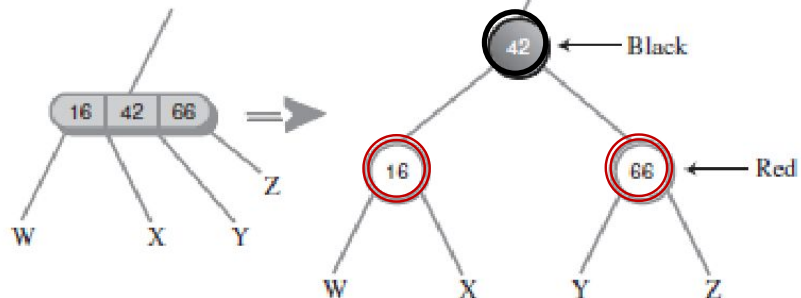
a) 2-Node



a) 3-Node



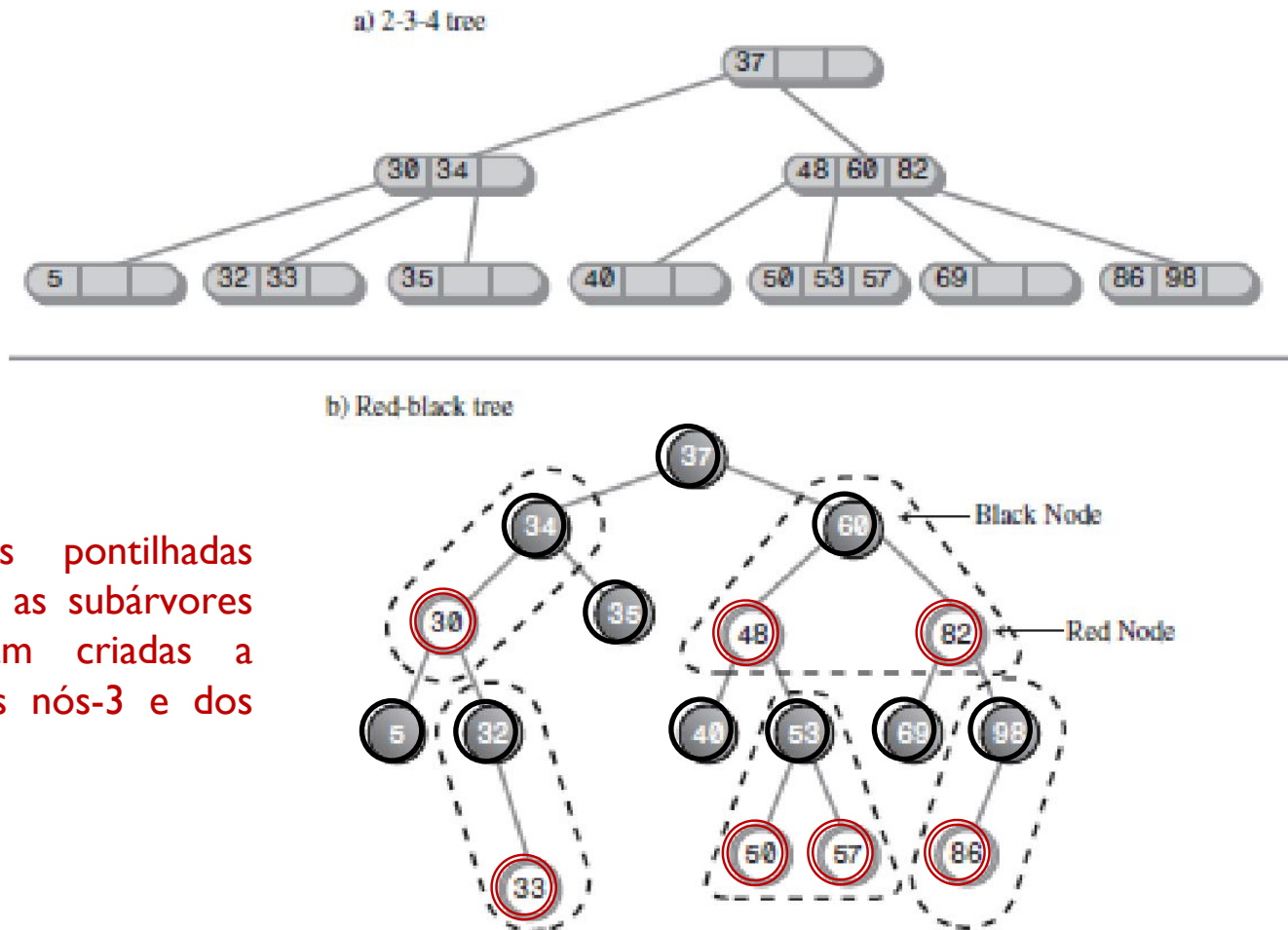
a) 4-Node



Árvores 2-3-4 e Vermelho-Preta

Exemplo - Transformação

- O exemplo da figura ilustra uma árvore 2-3-4 e a árvore vermelho-preta correspondente obtida aplicando-se as transformações.



Árvores 2-3-4 e Vermelho-Preta

Equivalência Operacional

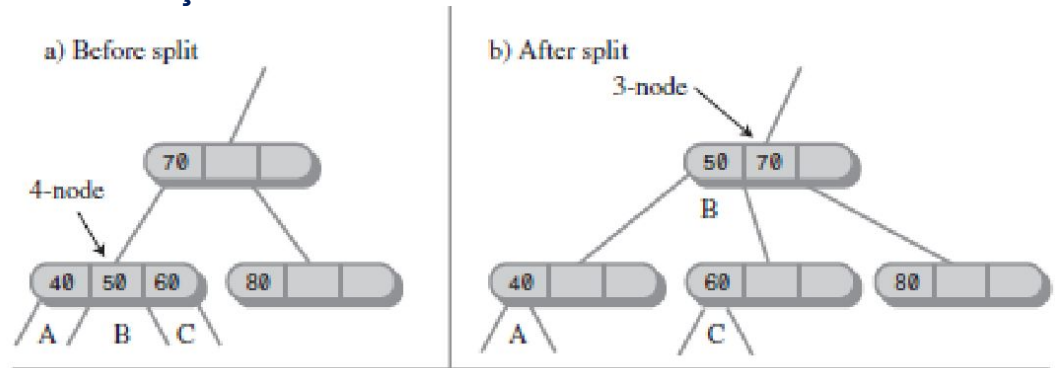
- Não só a estrutura de uma árvore vermelho-preta corresponde a de uma árvore 2-3-4, como as operações aplicadas nesses dois tipos de árvores para garantir o balanceamento são equivalentes.
- Em uma árvore 2-3-4 a árvore é mantida balanceada usando divisões de um nó. Em uma árvore vermelho-preta os dois métodos de balanceamento são mudanças de cor e rotações.

Árvores 2-3-4 e Vermelho-Preta

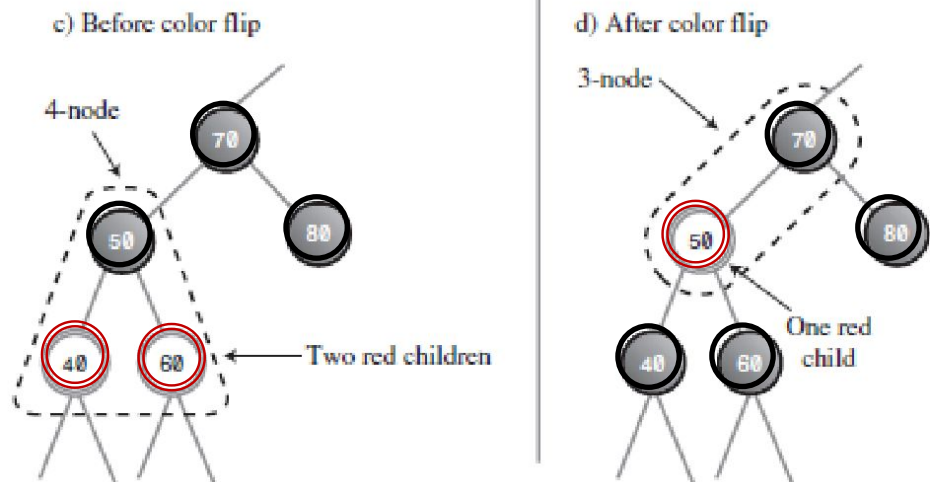
Equivalência Operacional em Nós-4

- A medida que o processo desce em uma árvore 2-3-4 buscando o ponto de inserção para um novo nó, divide cada nó-4 em dois nós-2. Na árvore vermelho-preta acontece uma mudança de cor.

- A figura a) mostra um nó-4 em uma árvore 2-3-4 antes dele ser dividido. A figura b) mostra o resultado após a divisão.



- A figura c) mostra a árvore vermelho-preta equivalente para a árvore 2-3-4 da figura a). A figura d) mostra o resultado após a mudança de cor.

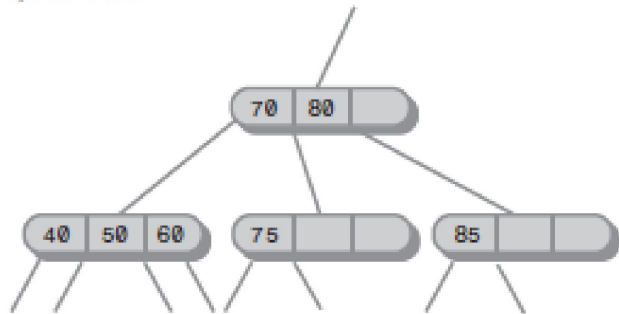


Árvores 2-3-4 e Vermelho-Preta

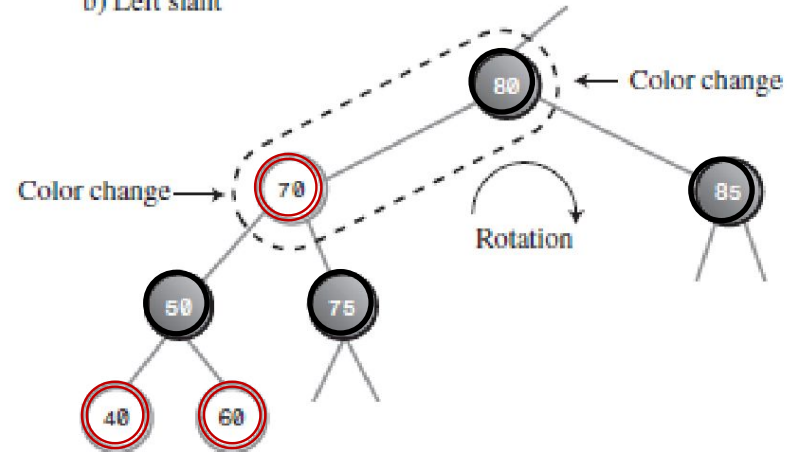
Equivalência Operacional em Nós-3

- A figura mostra a representação de uma árvore vermelho-preta no caso de um nó-3. Ilustra-se as rotações.

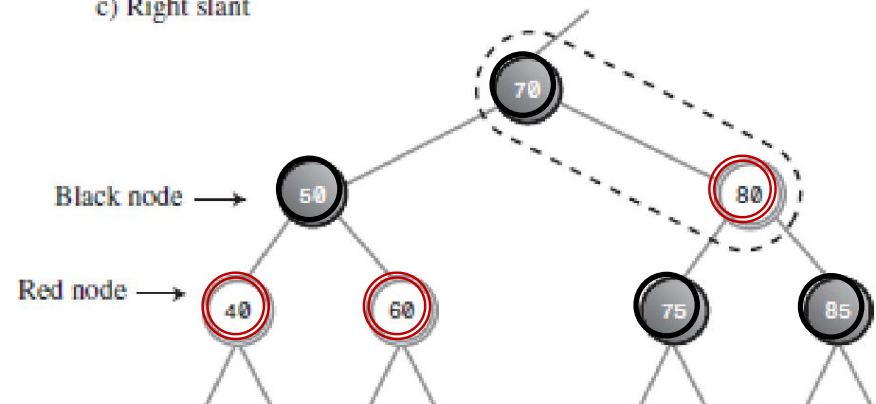
a) 2-3-4 tree



b) Left slant



c) Right slant



Árvores 2-3

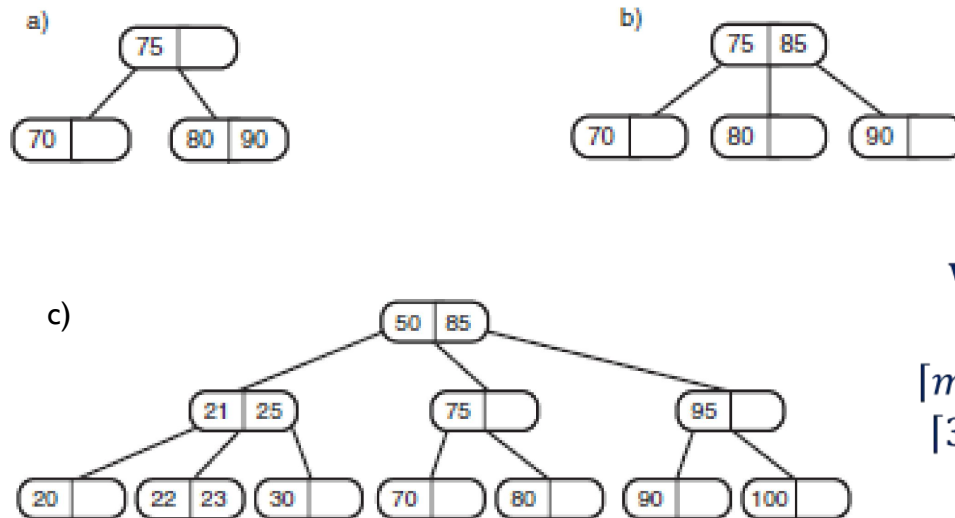
Definição

- Uma árvore 2-3 é um caso particular de árvore B, que de maneira semelhante as árvores 2-3-4, recebe seu nome pelo número de filhos que um nó pode ter.
- Neste tipo de árvore, um nó pode ter dois ou até três filhos. De acordo com isso, um nó pode armazenar uma chave (um elemento) ou até duas chaves (dois elementos).
- Historicamente este tipo de árvore é importante por ser a primeira árvore múltipla, inventada por J.E. Hopcroft em 1970. As árvores B foram inventadas posteriormente em 1972.
- Este tipo de árvore ainda é utilizada em muitas aplicações.
- Algumas técnicas usadas com árvores 2-3 são aplicáveis às árvores B.
- Vale observar que uma pequena mudança no número de filhos por nó pode causar uma grande alteração nos algoritmos de árvore.

Árvores 2-3

Exemplo

- Uma árvore 2-3 é um caso especial de árvore B de ordem 3.
- Observe que cada nó pode ter até 3 filhos. E que cada nó pode armazenar de um até dois elementos (ou chaves).
- Todos os nós folhas se encontram no mesmo nível.
- A organização dos valores-chave de um nó pai e de seus filhos é similar a das árvores 2-3-4, e das árvores B em geral.



Variação do número
de filhos:

$$\begin{aligned} \lfloor m/2 \rfloor &\leq k \leq m \\ \lfloor 3/2 \rfloor &\leq k \leq 3 \\ 2 &\leq k \leq 3 \end{aligned}$$

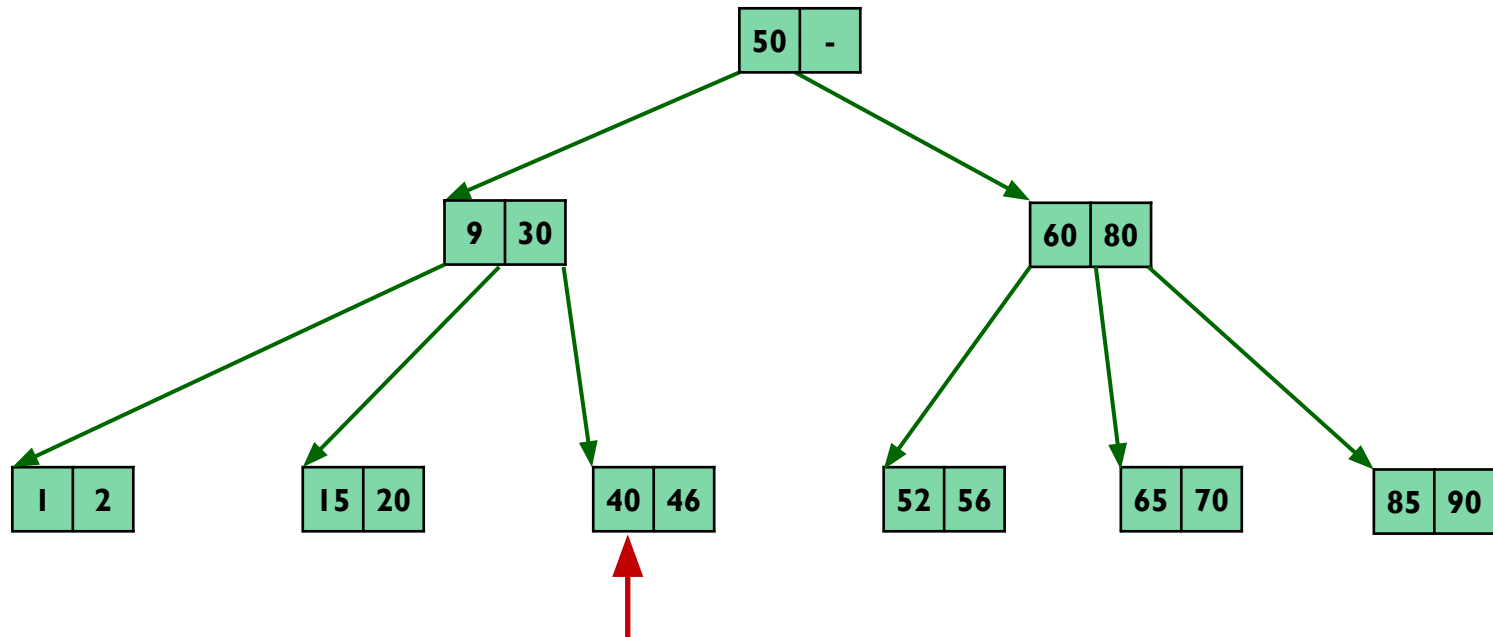
Variação do número
de entradas:

$$\begin{aligned} \lfloor m/2 \rfloor - 1 &\leq l \leq m - 1 \\ \lfloor 3/2 \rfloor - 1 &\leq l \leq 3 - 1 \\ 1 &\leq l \leq 2 \end{aligned}$$

Árvores 2-3

Busca

- A busca de um elemento é tratada exatamente como em uma árvore 2-3-4, exceto pelo número de elementos e filhos em cada nó, que é menor.
- No exemplo da figura, considere a busca do item com chave 40.



Árvores 2-3

Inserção

- Inserir um elemento em um nó é potencialmente simplificado porque menos comparações e movimentos são necessários. Como em árvores 2-3-4, toda inserção é feita em nós folhas.
- O processo de inserção é similar ao de uma árvore 2-3-4, mas há uma diferença importante no modo como as divisões são realizadas.

Árvores 2-3

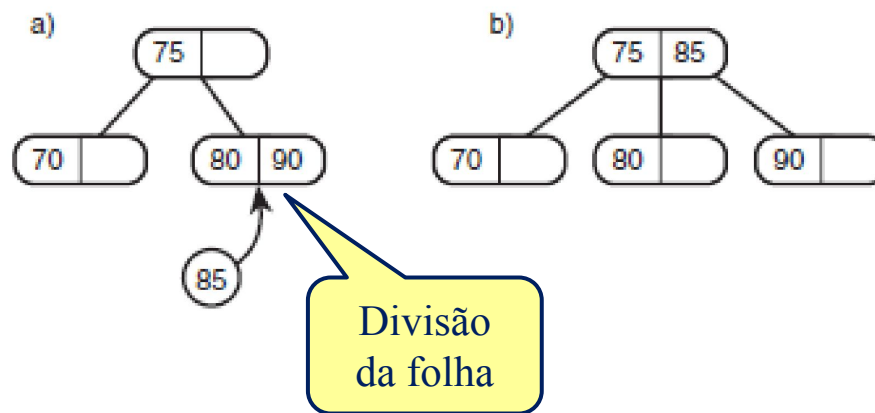
Inserção

- **Processo de Inserção:**
- O novo nó deve ser inserido em uma folha. Um nó completo em uma árvore 2-3 tem apenas **dois elementos**.
- Se o nó folha onde o novo item deve ser inserido não estiver completo, o novo elemento poderá ser inserido imediatamente. Porém, se o nó folha estiver **cheio**, ele terá que ser dividido. Os dois itens já existentes mais o novo item serão distribuídos entre três nós:
 - i) o nó folha original; ii) um novo nó, e iii) o nó pai.
- O menor item fica no nó original, o maior item é colocado no **novo nó**, enquanto o **elemento intermediário** é inserido no nó pai. Além disso, o novo nó deve ser conectado ao nó pai.
- Se o nó pai não estiver completo, o processo de inserção termina.
- Porém, se o nó pai estiver completo, o nó pai terá que ser **dividido também**. Este processo será propagado para cima até que um pai não completo ou a raiz seja encontrada. Caso a raiz estiver completa, uma nova raiz será criada.

Árvores 2-3

Inserção – Exemplo

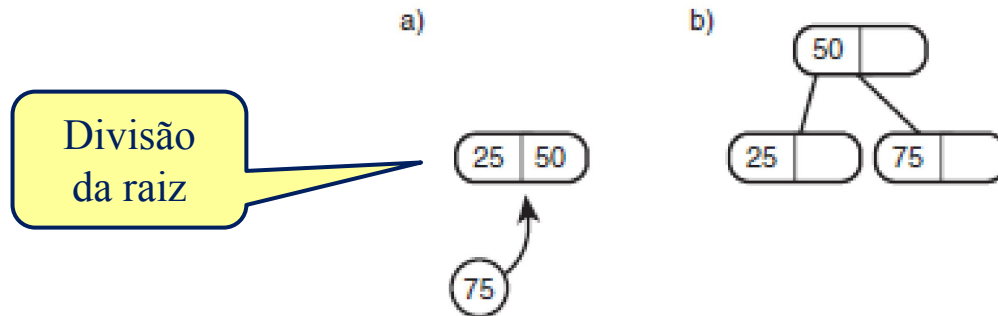
- A figura ilustra a inserção do novo item de dado com **chave 85** em uma folha completa. O nó folha é dividido e seus dados redistribuídos entre três nós: o próprio nó folha, um novo nó irmão e o nó pai. Como o nó pai desta folha não está completo, o processo de inserção termina após uma divisão.



Árvores 2-3

Inserção – Exemplo

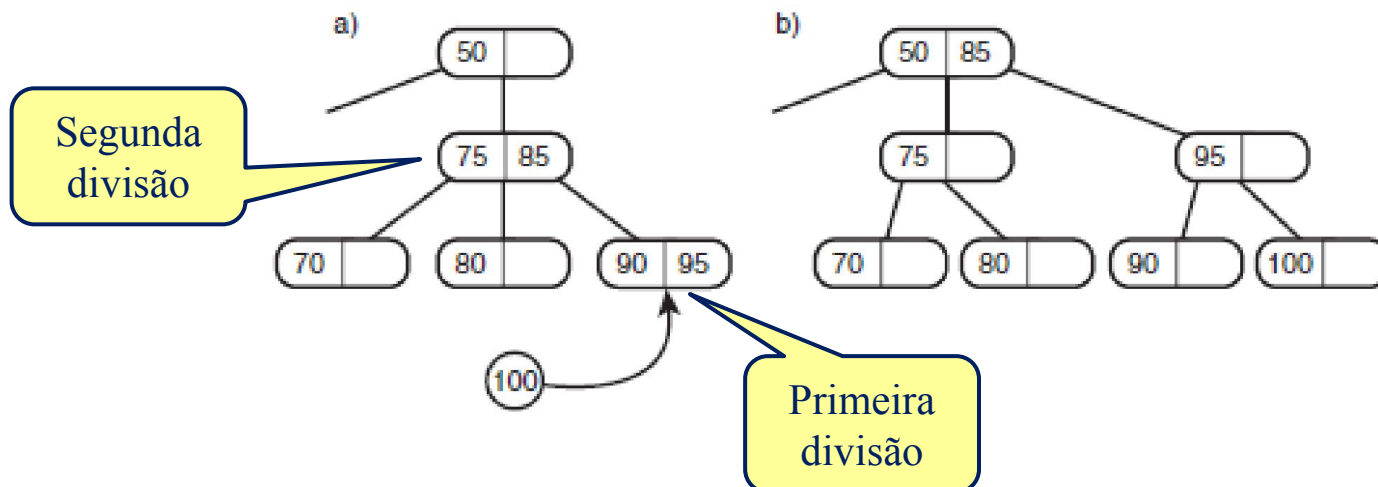
- A figura ilustra a inserção do novo item de dado com **chave 75** em um nó raiz completo. Como o nó raiz precisa ser dividido, neste caso dois novos nós precisam ser criados: um novo nó irmão da raiz original e um novo nó raiz.



Árvores 2-3

Inserção – Exemplo

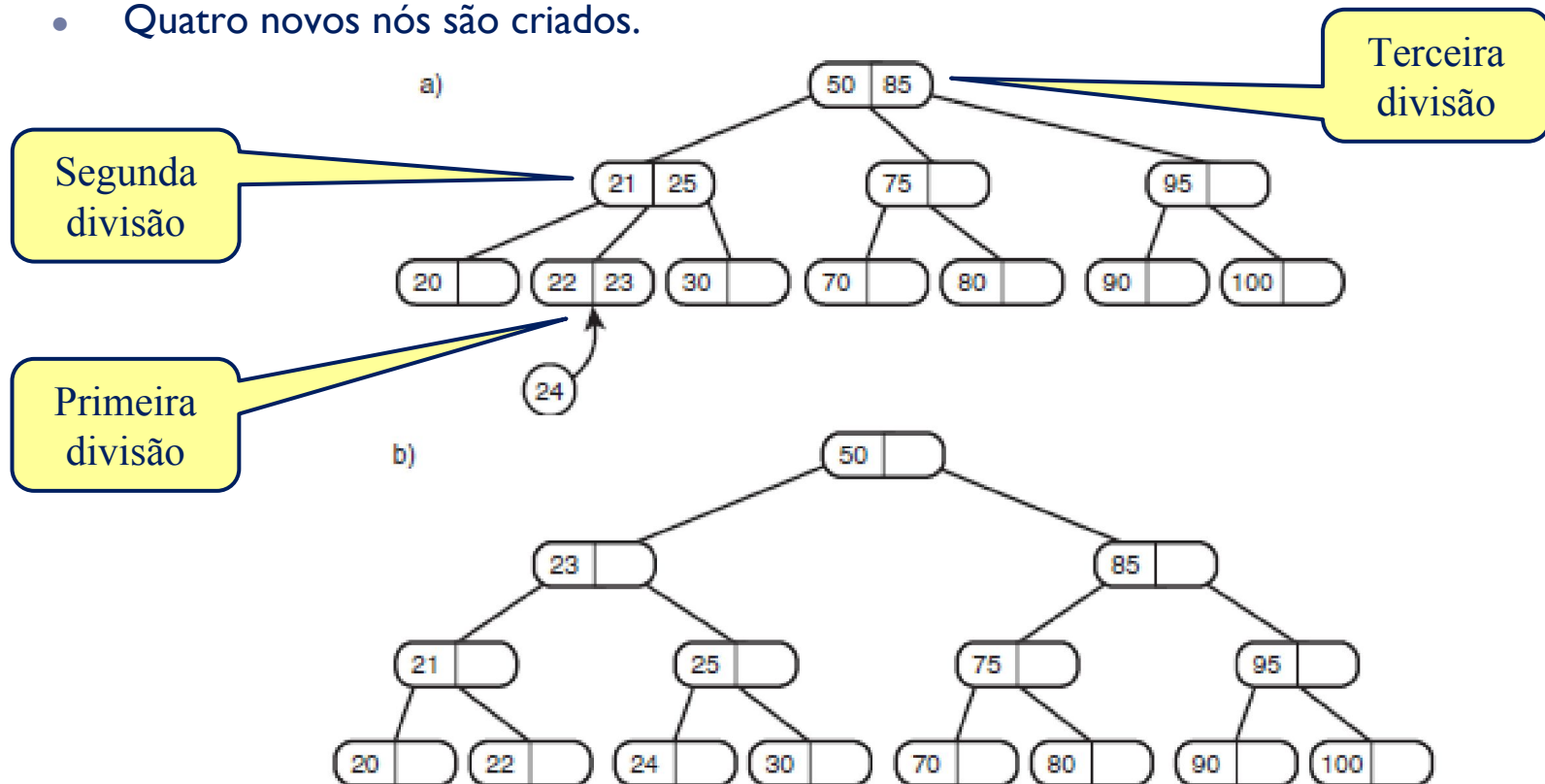
- A figura ilustra a inserção do novo item de dado com **chave 100** em uma folha completa. O nó folha é dividido e seus dados redistribuídos entre três nós: o próprio nó folha, um novo nó irmão e o nó pai. Como o nó pai também está completo, o processo de inserção continua. O nó pai também é dividido e seus dados redistribuídos entre três nós: o próprio nó pai, um novo nó irmão do pais e o nó pai do pai (o avô do nó folha).
- Como o nó pai do pai (o avô do nó folha) não está completo, o processo de inserção termina **após duas divisões**.



Árvores 2-3

Inserção – Exemplo

- A figura ilustra a inserção do novo item de dado com chave 24 em uma árvore 2-3. Neste exemplo, o processo de inserção propaga para cima uma série de divisões de nós até a raiz.
- Neste caso, são realizadas **três divisões de nós**, inclusive com a divisão do nó raiz.
- Quatro novos nós são criados.



Árvores 2-3

Inserção – Dicas de Implementação

- Para implementar a inserção de um novo nó precisamos lidar com divisões de nós.
- Inicialmente, a rotina de inserção começa pela raiz e pesquisará a árvore descendo até encontrar a folha apropriada. Se a folha não estiver completa, inserirá o valor. Porém se a folha estiver completa, ele terá que reorganizar a árvore realizando divisões de nós.
- Para isso chamará uma função **split()**. Esta função receberá o nó a ser dividido e o novo item a ser inserido. Se **split()** encontrar nó pai completo, chamará a si mesmo recursivamente para dividir o pai. Continuará chamando a si mesmo até que o nó pai não estiver completo ou encontrar a raiz.
- O retorno da função **split()** será o novo nó à direita que será usado pela chamada recursiva anterior do **split()**.

Árvores 2-3

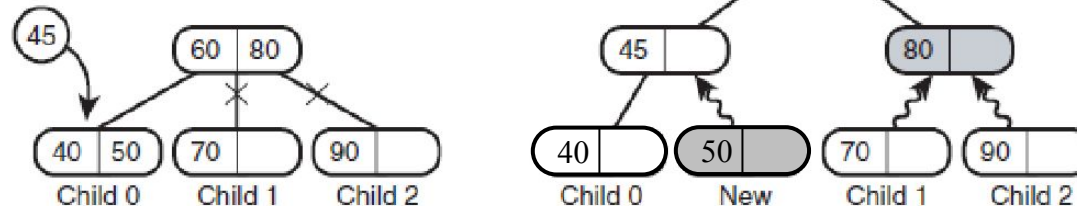
Inserção – Dicas de Implementação

- O processo de divisão de um nó pai, criará um segundo pai. Teremos um nó pai à esquerda e um novo pai à direita.
- Precisaremos alterar as conexões iniciais de um único pai com três filhos para as conexões finais de dois pais com dois filhos.
- Existem três casos possíveis dependendo de qual filho está sendo dividido.
- Os novos nós criados como resultado da divisão são mostrados sombreados e as novas conexões são mostradas como linhas onduladas.

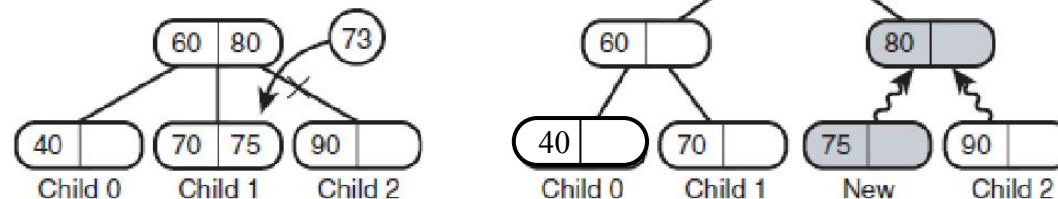
Árvores 2-3

Inserção – Dicas de Implementação

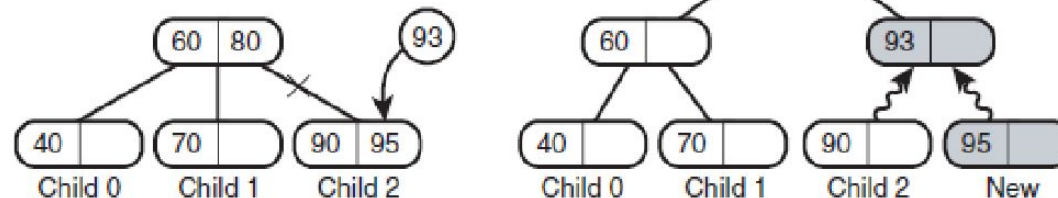
a) Splitting Child 0



b) Splitting Child 1



c) Splitting Child 2



Referências

- Adam **Drodzek**. Estrutura de dados e Algoritmos em C++. Cap7. 2002. Cengage Learning.
- Robert **Lafore**. Estrutura de dados e Algoritmos em Java. Cap10. 2003. Ciência Moderna.