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

Árvore Múltiplas
Árvores B

Disciplina: Estrutura de Dados II

Prof. Fermín Alfredo Tang Montané

Curso: Ciência da Computação

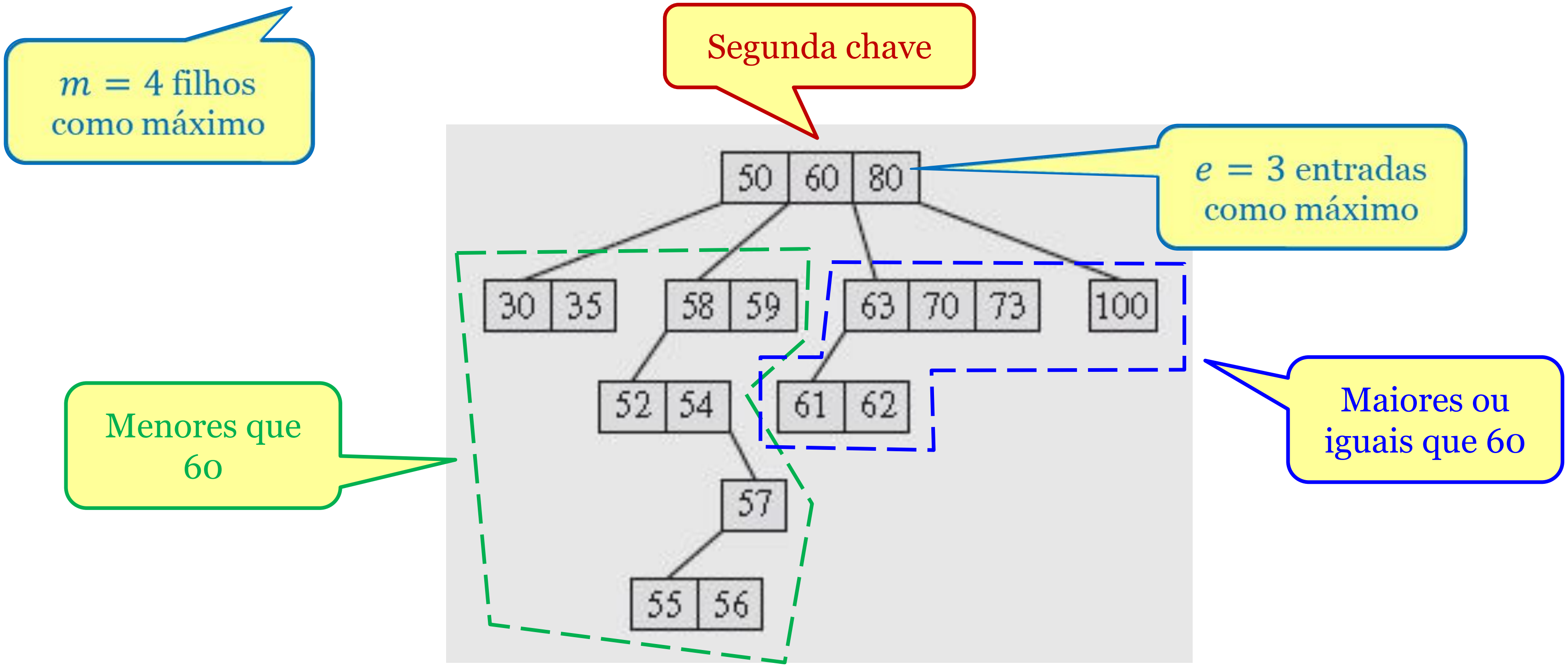
Árvores Múltiplas
Definição

- Uma árvore múltipla de ordem *m* (árvore *m*-ária) é uma árvore de busca na qual se atendem as seguintes condições:
  1. cada nó tem até *m* filhos (subárvores) e até *m-1* entradas (ou chaves);
  2. em cada nó, as chaves estão em ordem ascendente;
  3. em cada nó, a i-ésima chave é maior que as chaves dos primeiros *i* filhos;
  4. em cada nó, a i-ésima chave é menor ou igual que as chaves dos filhos (i+1)-ésimo até o último.

- **Obs.** As árvores múltiplas ou (árvore *m*-ária) também são conhecidas como *multiway trees* (árvores multi-caminhos).

Árvores Múltiplas
Exemplo

- Exemplo de uma árvore múltipla de ordem 4, ou árvore quaternária (*4-way tree*).



Árvores Múltiplas
Deficiência

- As árvores de busca *m*-árias desempenham o mesmo papel das árvores binárias de busca: rápida recuperação e atualização de informação.
- As deficiências são semelhantes. Necessidade de balanceamento a medida que elementos são inseridos e removidos. No entanto a estrutura possui mecanismos diferenciados de balanceamento.

Árvores Múltiplas
Introdução

- Em muitas aplicações, uma tabela pode ser considerada muito grande.
  - As chaves não podem ser mantidas todas na memória principal
- Assim, a tabela precisa ser mantida em memória secundária
  - Alto custo de acesso
- Necessidade de uma estrutura que minimize o tempo de acesso em memória secundária para:
  - Buscas
  - Inserções
  - Remoções

Árvore B
Introdução

- As árvores B foram propostas por Bayer e MacCreight (1972).
- As árvores B permitem manter várias chaves em cada nó da estrutura (muitas vezes chamadas de página).
- Proporciona uma organização de ponteiros de forma que as operações são executadas rapidamente.
- Sua construção assegura que todas as folhas se encontram no mesmo nível, não importando a ordem de entrada dos dados. Sendo por tanto uma estrutura balanceada.
- Largamente utilizadas como forma de armazenamento em memória secundária.
- Diversos sistemas comerciais de banco de dados utilizam árvores B.

Árvore B
Introdução

- O tamanho de cada nó pode ser tão grande quanto o tamanho de um bloco do disco.
- O numero de chaves de um nó pode variar dependendo do:
  - Tamanho das chaves;
  - Tamanho de um bloco (alguns autores chamam de pagina);
- Tamanho do bloco varia para cada sistema 512 bytes, 4K ou mais.

Árvores B
Definição

- Uma árvore B de ordem m é uma árvore de busca múltipla (ou multi-caminho), 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
Definições alternativas

- Definições alternativas definem:
  - A ordem m da árvore B como o número mínimo de filhos de um nó, onde cada nó possui $l = k - 1$ entradas e k filhos, onde $m \leq k \leq 2m$.
  - A ordem d da árvore B como o número mínimo de entradas de um nó, onde cada nó possui l entradas e $k = l + 1$ filhos, onde $d \leq l \leq 2d$.

- De acordo com a esta última definição, em uma árvore B de ordem d:
  - O nó raiz possui um número de entradas l (chaves), onde: $1 \leq l \leq 2d$.
  - Cada nó não-raiz possui um número de entradas l, onde: $d \leq l \leq 2d$.
  - O nó raiz pode ser uma folha, ou ter no mínimo 2 filhos e no máximo $2d+1$ filhos.
  - Cada nó interno possui no mínimo $d+1$ filhos e no máximo $2d+1$ filhos.
  - Em cada nó as chaves estão ordenadas.

Árvore B
Exemplo 1

- Ilustrando casos extremos para um **nó raiz sem filhos**, em uma árvore de ordem m=5.

$m = 5$ filhos como máximo

Raiz sem filhos

50	-	-	-

$e = 4$ entradas como máximo

Número de entradas:
$l = 1$

Número de filhos na raiz:
$k = 0$

Raiz sem filhos

50	70	80	95

Número de entradas:
$l = 4$

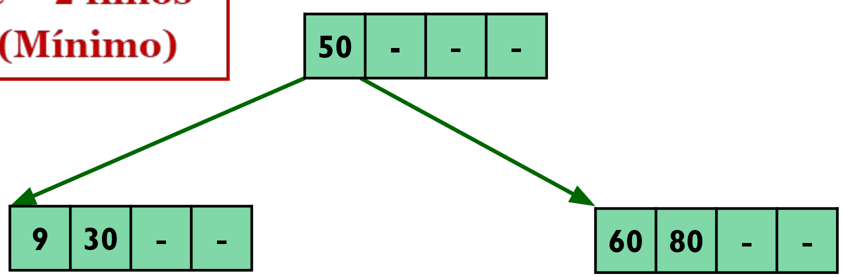
Variação do número de entradas na raiz:

$$1 \leq l \leq m - 1$$
$$1 \leq l \leq 5 - 1$$
$$1 \leq l \leq 4$$

Árvore B
Exemplo 2

- Ilustrando casos extremos para um **nó raiz com filhos**, em uma árvore de ordem m=5

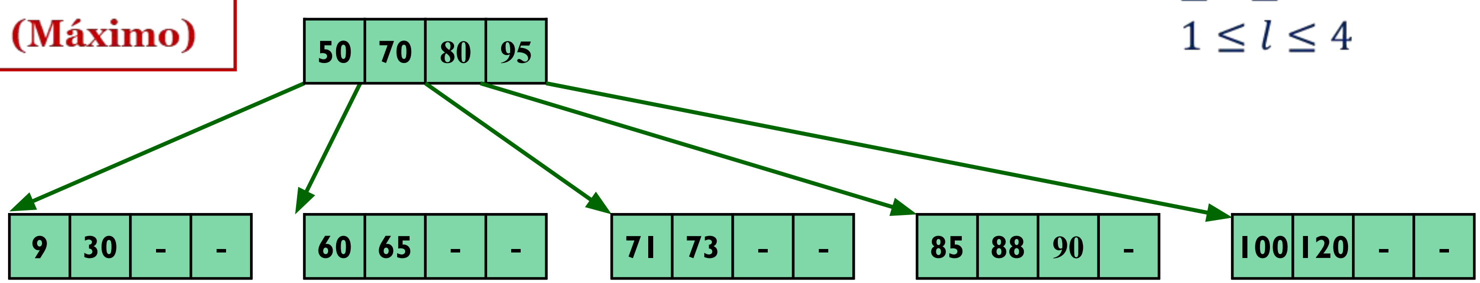
Raiz com $k = 2$ filhos (Mínimo)



Variação do número de filhos na raiz:

$$2 \leq k \leq m$$
$$2 \leq k \leq 5$$

Raiz com $k = 5$ filhos (Máximo)



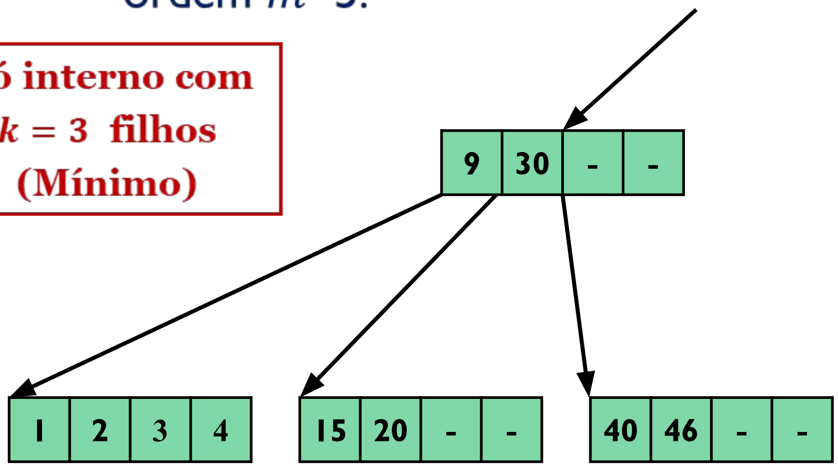
Variação do número de entradas na raiz:

$$1 \leq l \leq m - 1$$
$$1 \leq l \leq 5 - 1$$
$$1 \leq l \leq 4$$

Árvore B
Exemplo 3

- Ilustrando casos extremos para qualquer **nó interno**, em uma árvore de ordem m=5.

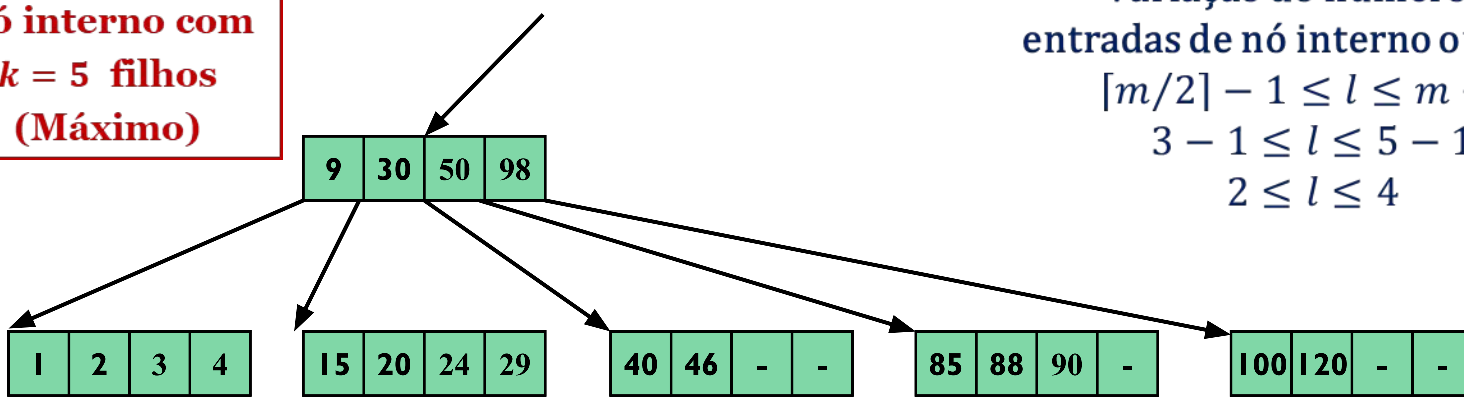
Nó interno com k = 3 filhos (Mínimo)



Variação do número de filhos de um nó interno:

$$\lceil m/2 \rceil \leq k \leq m$$
$$\lceil 5/2 \rceil \leq k \leq 5$$
$$3 \leq k \leq 5$$

Nó interno com k = 5 filhos (Máximo)



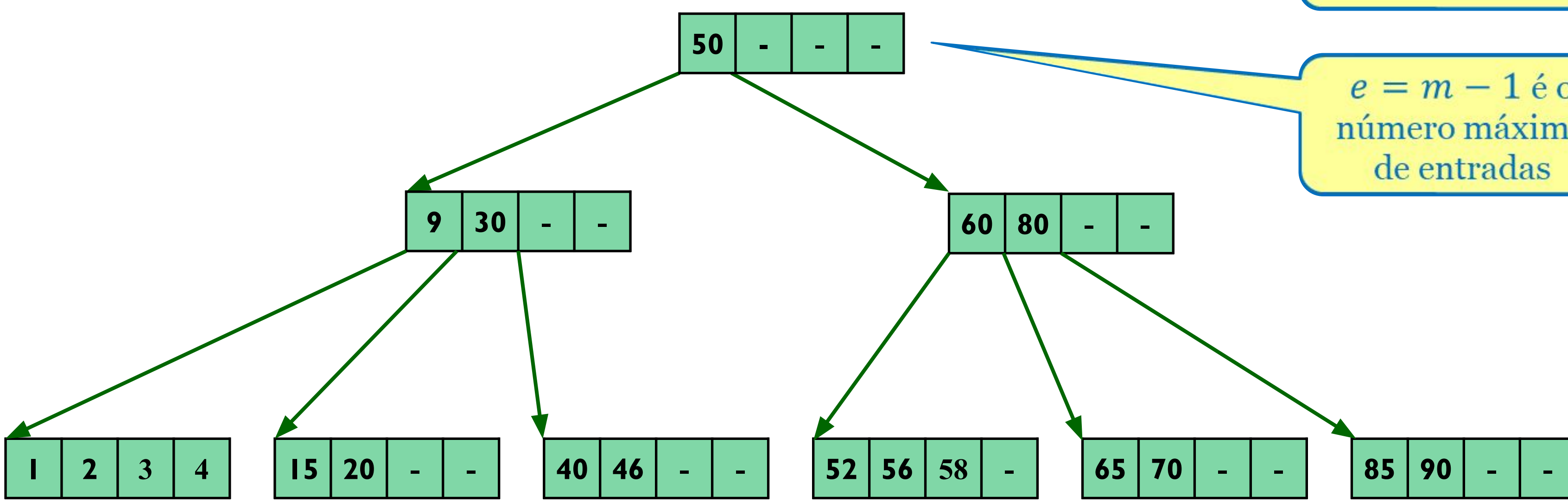
Variação do número de entradas de nó interno ou folha:

$$\lceil m/2 \rceil - 1 \leq l \leq m - 1$$
$$3 - 1 \leq l \leq 5 - 1$$
$$2 \leq l \leq 4$$

Árvore B
Exemplo 3

- Ilustrando uma árvore B completa de ordem m=5.

m é o número máximo de filhos



$e = m - 1$ é o número máximo de entradas

Variação do número de filhos de um nó interno:

$$\lceil m/2 \rceil \leq k \leq m$$
$$\lceil 5/2 \rceil \leq k \leq 5$$
$$3 \leq k \leq 5$$

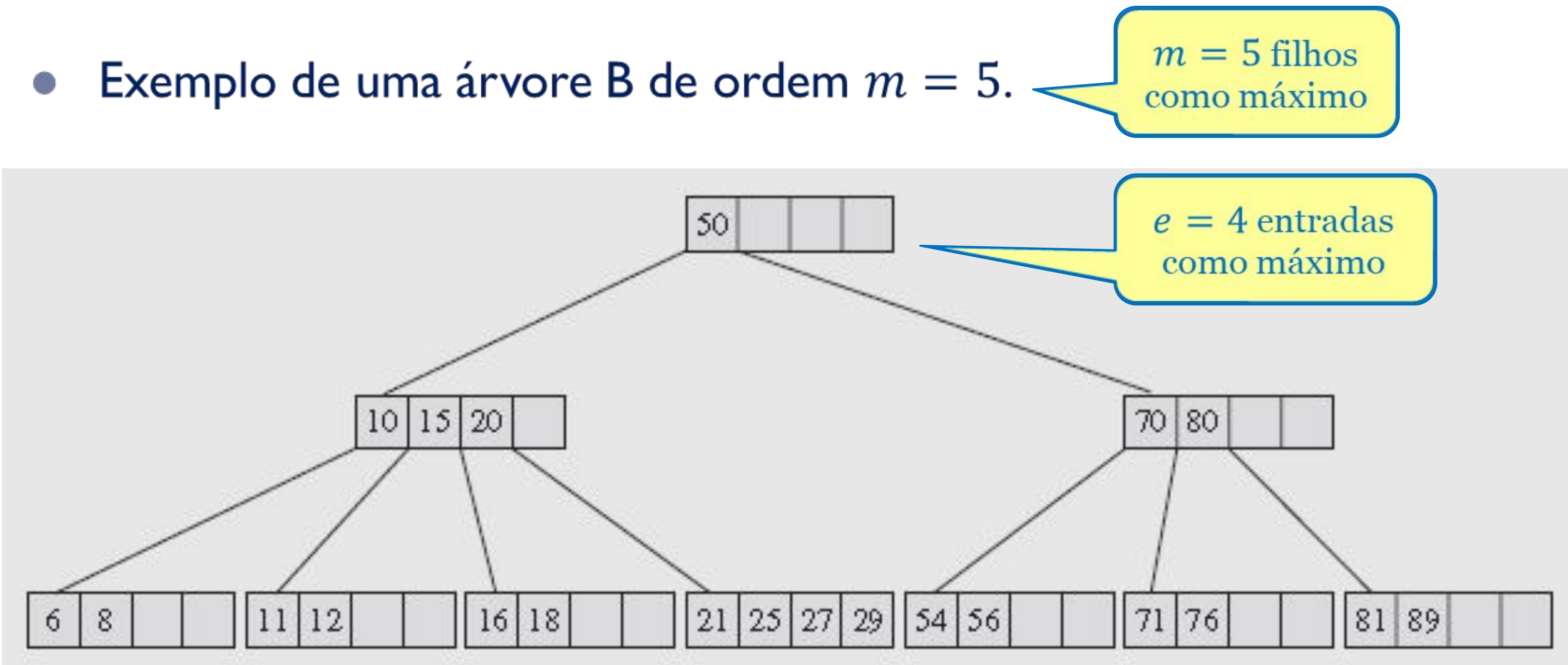
Variação do número de entradas de nó interno ou folha:

$$\lceil m/2 \rceil - 1 \leq l \leq m - 1$$
$$3 - 1 \leq l \leq 5 - 1$$
$$2 \leq l \leq 4$$

$$\boxed{k = l + 1}$$

Árvores B
Exemplo 4

- Exemplo de uma árvore B de ordem $m = 5$.



Variação do número de filhos de um nó interno:

$$\lceil m/2 \rceil \leq k \leq m$$
$$3 \leq k \leq 5$$

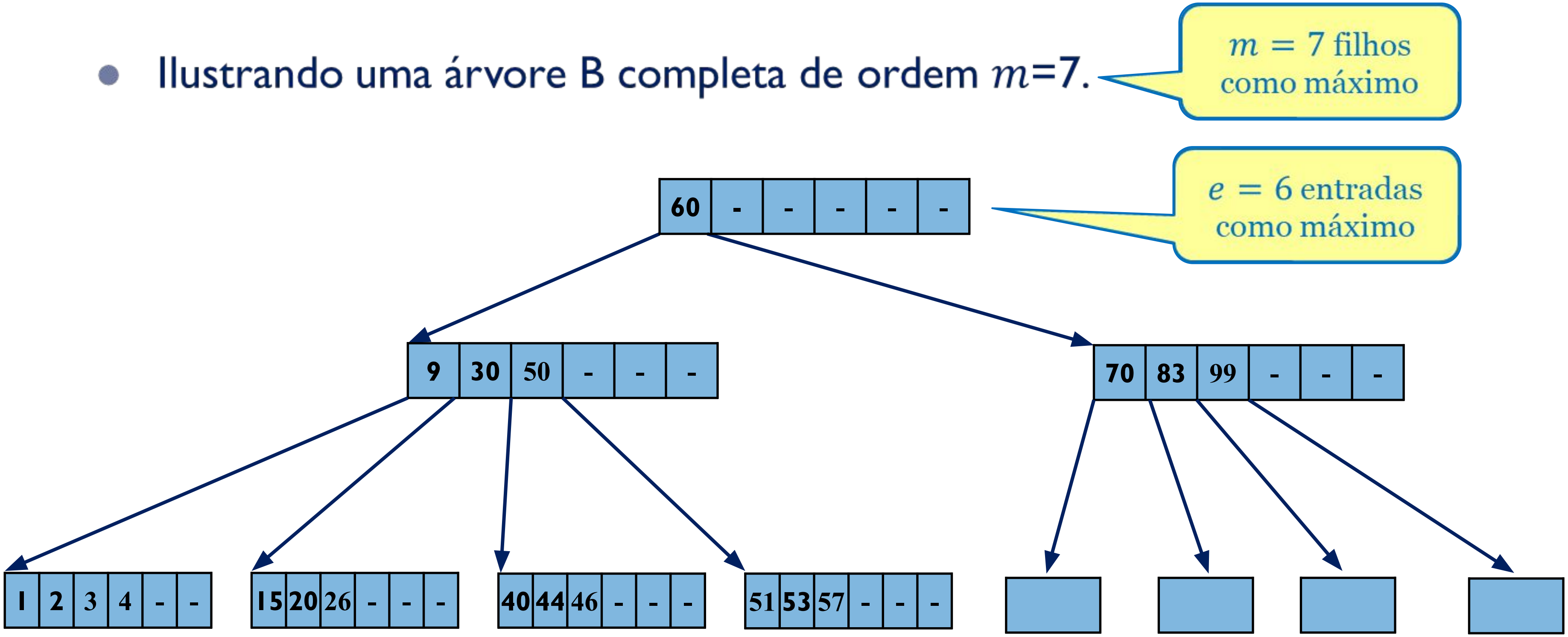
Variação do número de entradas de nó interno ou folha:

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

$$\boxed{k = l + 1}$$

Árvore B
Exemplo 5

- Ilustrando uma árvore B completa de ordem m=7.



Variação do número de filhos de um nó interno:

$$\lceil m/2 \rceil \leq k \leq m$$
$$\lceil 7/2 \rceil \leq k \leq 7$$
$$4 \leq k \leq 7$$

Variação do número de entradas de nó interno ou folha:

$$\lceil m/2 \rceil - 1 \leq l \leq m - 1$$
$$4 - 1 \leq l \leq 7 - 1$$
$$3 \leq l \leq 6$$

$$\boxed{k = l + 1}$$

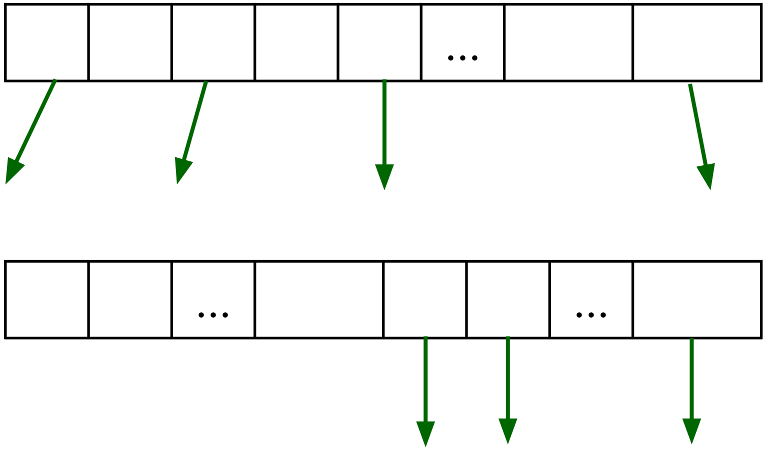
Árvores B
Comportamento

- As árvores B, de acordo com a sua definição oferecem a garantia de estar sempre 50% cheias com relação ao espaço total alocado para elas, o que significa que no pior caso, 50% do espaço alocado pode ser desperdiçado. Mas com que frequência isso acontece?
- Análises e simulações, indicam que depois de uma série de numerosas inserções e remoções aleatórias, a árvore B fica 69% cheia (Yao, 1978), e que depois disso as mudanças na porcentagens de células ocupadas são muito pequenas.

Árvore B
Representação

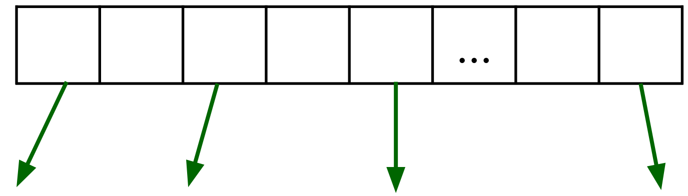
- A representação de uma árvore B ordem m, pode ser realizada definindo uma estrutura contendo:
  - Um vetor para armazenar um máximo de $m-1$ entradas (chaves).
  - Um vetor para armazenar os ponteiros para um máximo de m nós filhos.
  - Um inteiro para registrar o número k' de entradas presentes no nó.
  - Um booleano para indicar se o nó é uma folha ou nó interno.

- Onde:
  - k_i representa as entradas, $\mathrm{i} = 1, \ldots, m-1$.
  - p_i representa os ponteiros aos nós filhos, $\mathrm{i} = 0, \ldots, m-1$.



Árvore B
Representação

- Considere uma árvore de ordem m, e um nó interno com l entradas, onde $(\lceil m/2 \rceil - 1) \leq l \leq (m-1)$:
  - As entradas desse nó, são denotadas como: $k_1, \ldots, k_l$;
  - Os ponteiros para os filhos desse nó, são denotadas como: $p_0, \ldots, p_l$.



- **Propriedade:**
- Para qualquer chave y que pertence ao nó apontado pelo ponteiro p_0, o valor da chave $y < k_1$.
- Para qualquer chave y que pertence ao nó apontado pelo ponteiro p_l, o valor da chave $y > k_l$.
- Para qualquer chave y que pertence ao nó apontado pelo ponteiro p_i, o valor da chave $k_i < y < k_{i+1}$, onde $1 \leq i \leq l$.

Árvore B
Busca

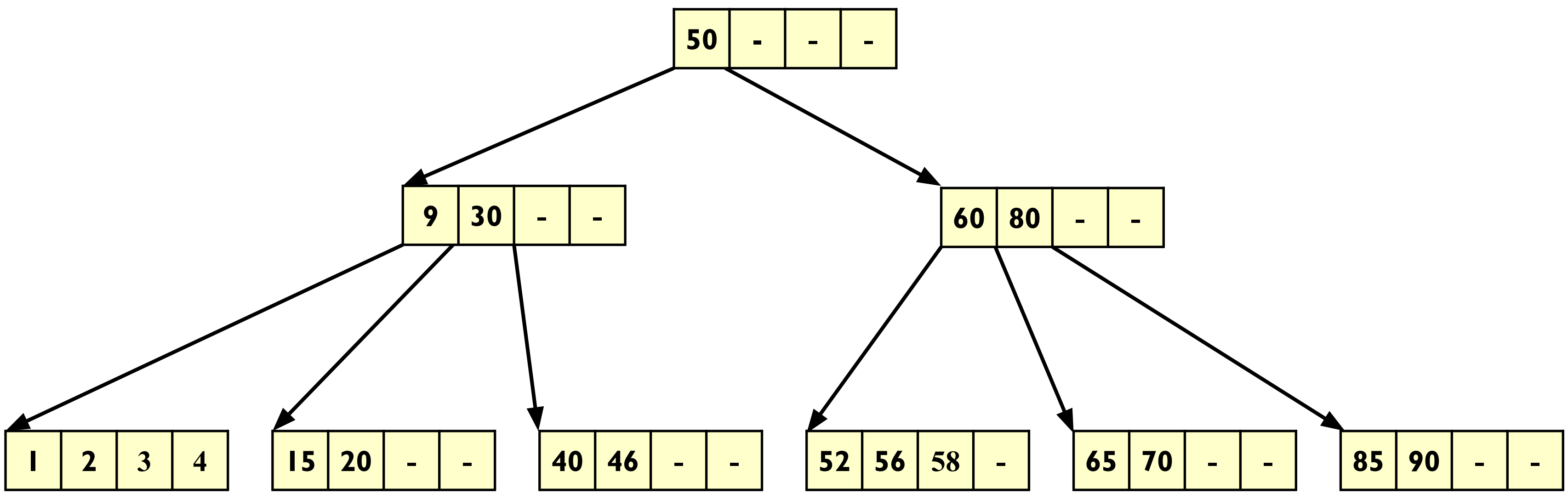
- Buscar uma chave y em uma árvore B.
- Processo de busca semelhante ao utilizado para árvore de busca binária:
  - Acrescenta-se testes relativos às chaves existentes de cada nó.
  - Realiza-se um busca sequencial (ou binária) dentro de um nó.

Árvore B
Algoritmo de busca

- Buscar uma chave y em uma árvore B.
- A busca começa no nó raiz, denotado como n_0. Em cada iteração, examina-se um determinado nó, denotado de forma genérica como n_j.
- O algoritmo da busca compara a chave y, a chave procurada, com as chaves do nó n_j.
- Se a chave não se encontra no nó n_j, a busca prossegue em um nó n_{j+1} filho do nó anterior, de acordo com a propriedade:
  - Se o valor da chave $(y < k_1)$ acessar o nó apontado pelo ponteiro p_0.
  - Se o valor da chave $(k_i < y < k_{i+1})$ acessar o nó apontado pelo ponteiro p_i, onde $1 \leq i \leq l$.
  - Se o valor da chave $(y > k_l)$ acessar o nó apontado pelo ponteiro p_l.
- A busca termina com sucesso se a chave y é encontrada no novo nó.
- A busca termina sem sucesso ao atingir-se um nó folha que não contém a chave procurada.

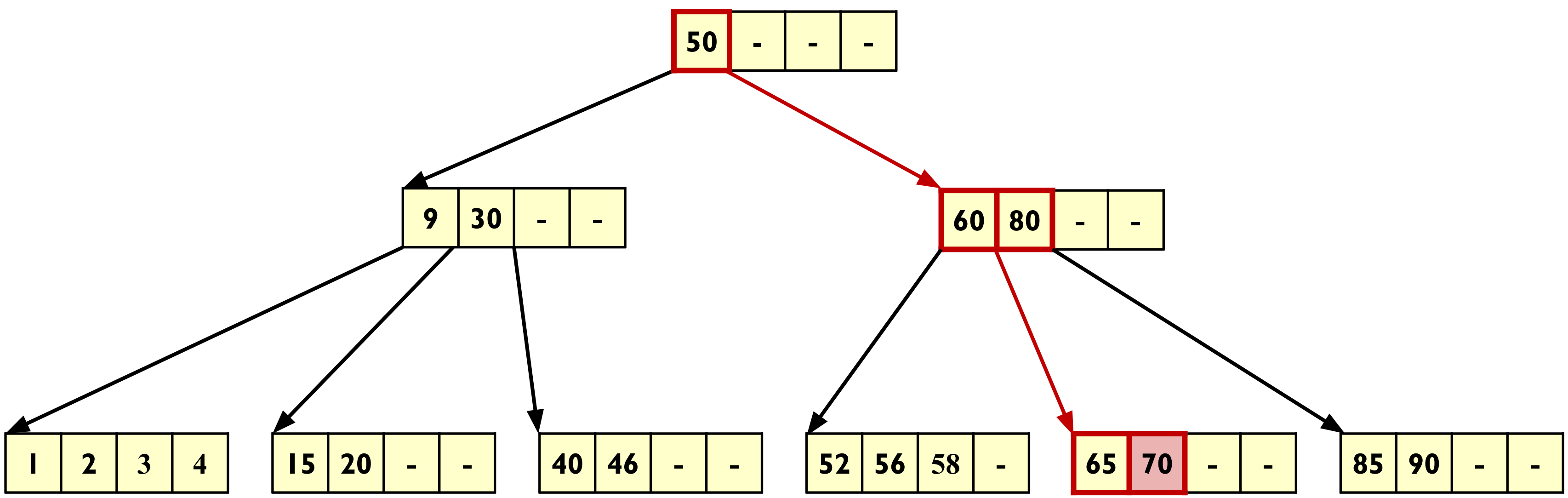
Árvore B
Busca

- Ilustra-se a busca da chave y =70 na árvore B com ordem m=5.



Árvore B
Busca

- Ilustra-se a busca da chave y=70 na árvore B com ordem m=5.



- Busca terminada com sucesso!.

Árvore B
Inserção

- A inserção de um novo elemento em uma árvore B de ordem m, somente será válida se a chave y do novo elemento não existe.
- O primeiro passo da inserção consiste na busca da posição de inserção para o novo elemento. Chamaremos a este nó de nó destino P.
- Um problema que deve ser tratado ao se inserir um novo elemento em um nó é a possibilidade de acontecer um ***overflow*** no nó destino da inserção. Isto significa que o nó destino P, já possui o número máximo de chaves permitidas $(m - 1)$.
- Neste caso, inserir um novo nó produz um *overflow* (m chaves) o que deve ser tratado mediante a cisão (divisão) do nó destino em dois.

Árvore B
Cisão de um nó

- Seja um nó destino P com *overflow* (m chaves).
- Para simplificar consideraremos m impar, o que significa que $m - 1$ par.
- Consiste em dividir o nó destino P, em 2 nós com $(\lceil m/2 \rceil - 1)$ chaves cada.
- A divisão do nó consiste em:
  - Identificar a chave central k_i do nó P.
  - Criar um novo nó Q com as chaves maiores que k_i.
  - Eliminar o excesso de nós em P, que fica com as chaves menores que k_i.
  - A chave central k_i é inserida no nó W, pai do nó P, juntamente com um ponteiro para o novo nó Q.
- A cisão de um nó é propagável é pode atingir a raiz.

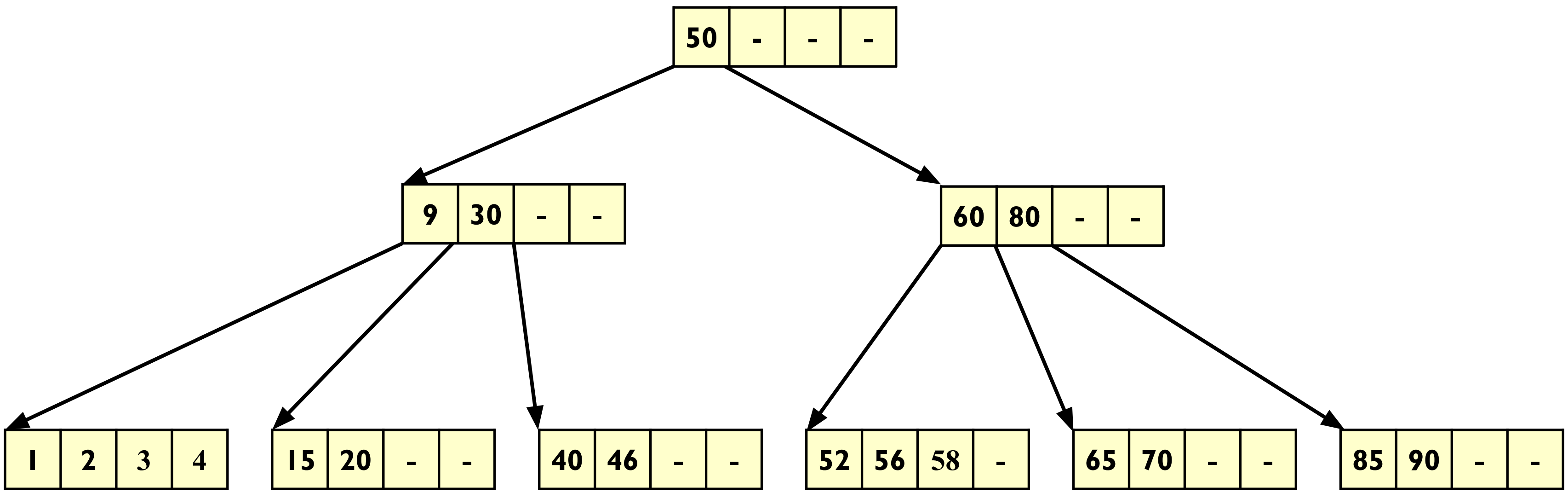
- **Observação:** No caso em que m é par, a divisão do nó P não será exata, uma metade terá mais elementos que a outra. No entanto deve ser garantido que uma das metades possua $(m/2 - 1)$ enquanto a outra possui $m/2$.

Árvore B
Algoritmo de Inserção

- Inserir uma chave y em uma árvore B
- Passo 1: aplicar procedimento Busca, verificando a validade da inserção (a inserção é válida somente se a chave y não existe).
- Passo 2: se a inserção é válida, incluir o novo elemento na i-ésima posição do nó P.
- Passo 3: verificar se o nó P necessita de cisão. Em caso afirmativo, propagar a cisão enquanto for necessário.

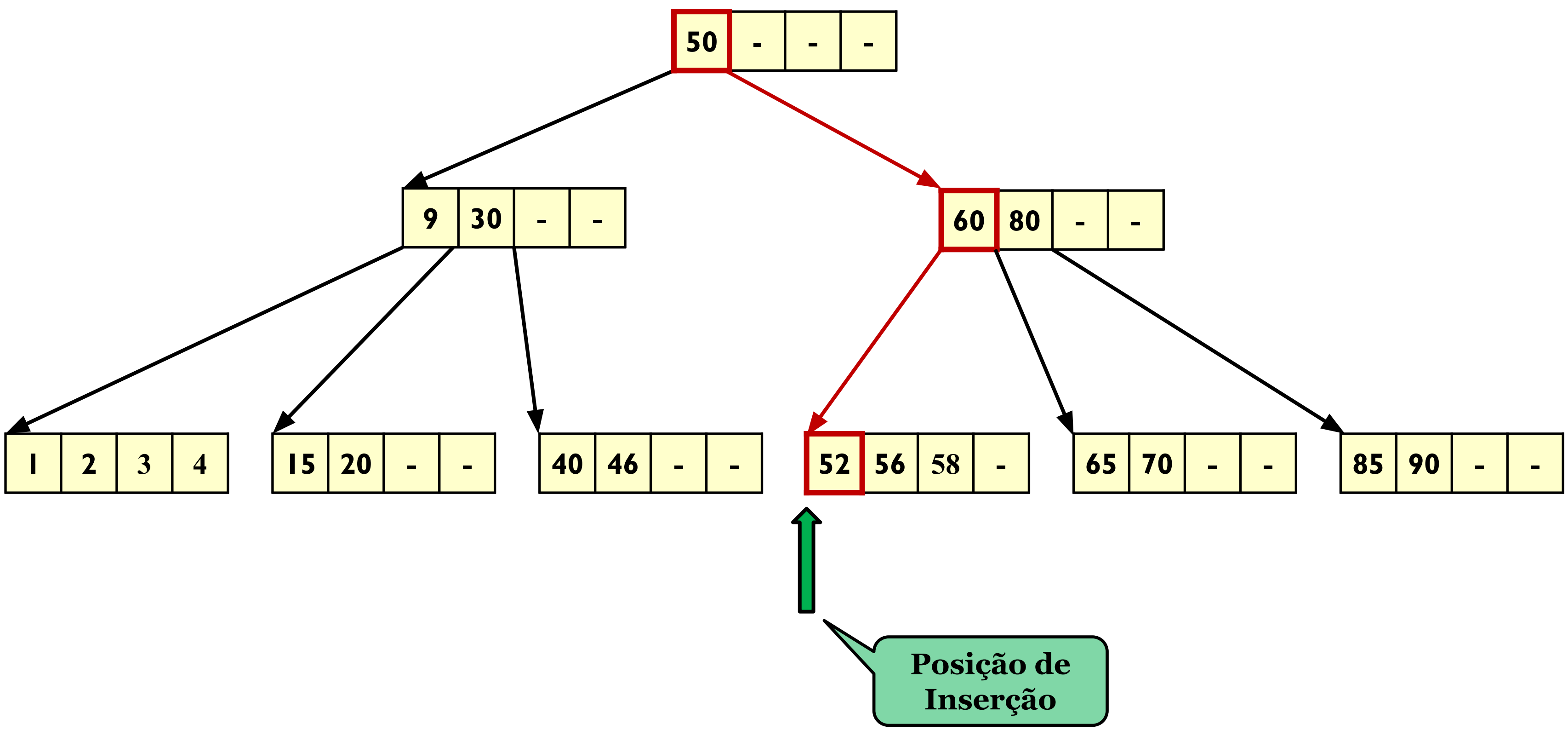
Árvore B
Inserção

- Ilustra-se a inserção da chave y=51 na árvore B com ordem m=5.



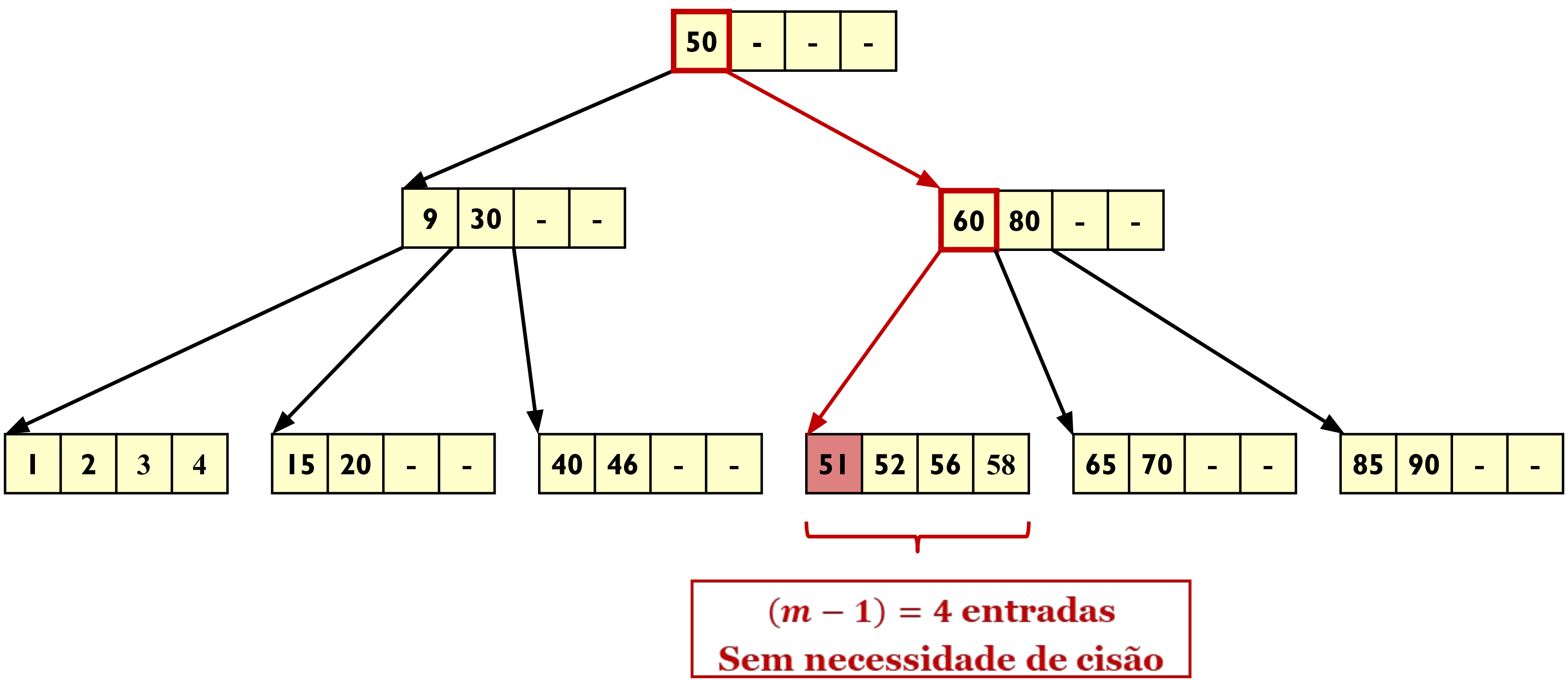
Árvore B
Inserção

- Ilustra-se a inserção da chave y=51 na árvore B com com ordem m=5.



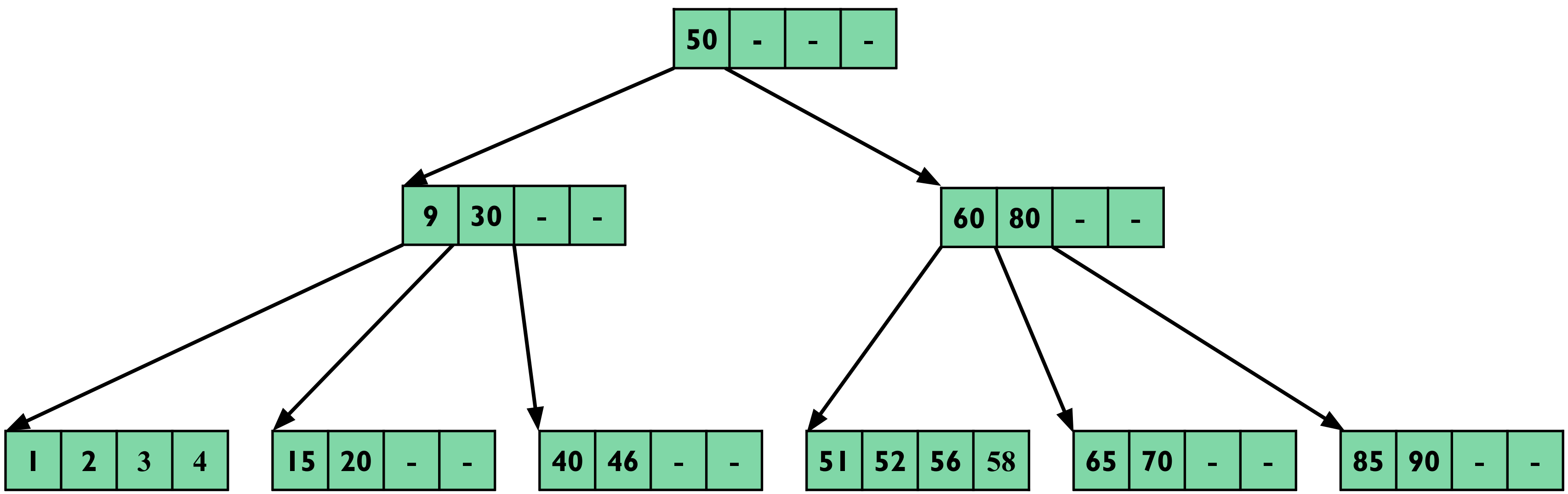
Árvore B
Inserção

- Ilustra-se a inserção da chave y=51 na árvore B com ordem m=5.



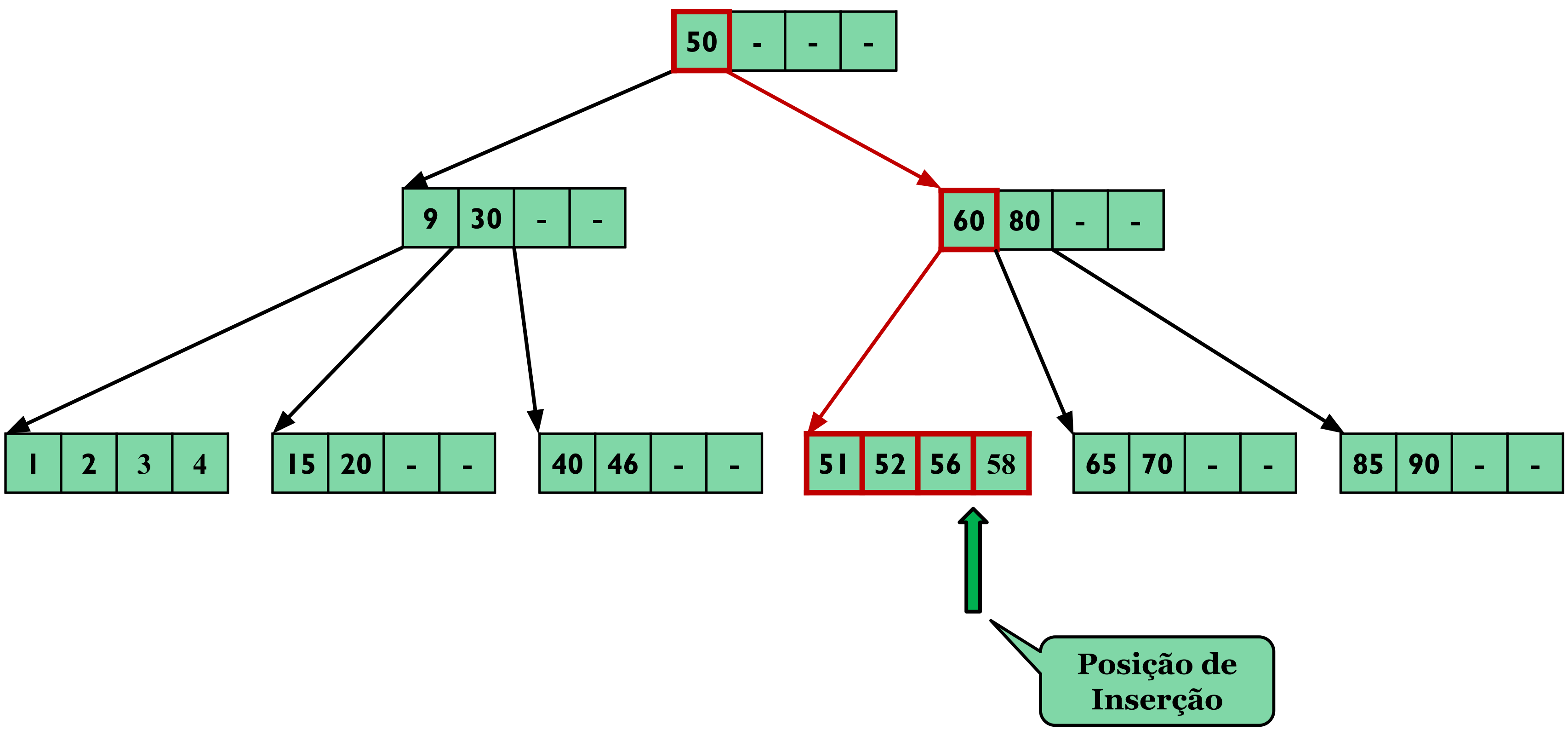
Árvore B
Inserção

- Ilustra-se a inserção da chave y=57 na árvore B com ordem m=5.



Árvore B
Inserção

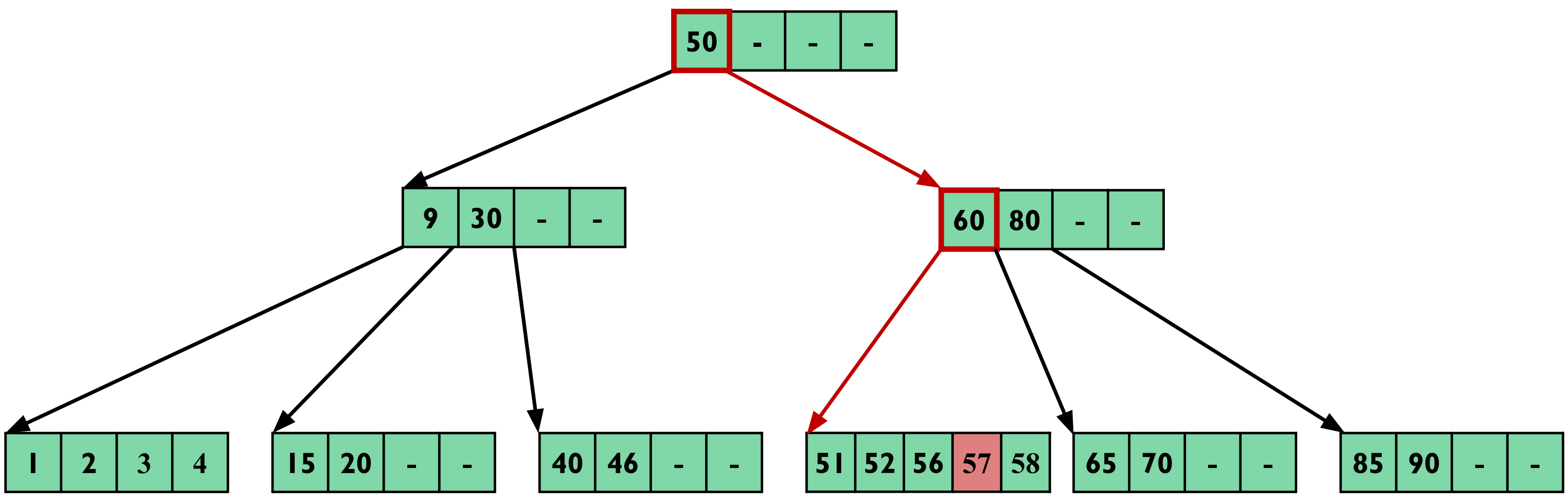
- Ilustra-se a inserção da chave y=57 na árvore B com ordem m=5.



Árvore B

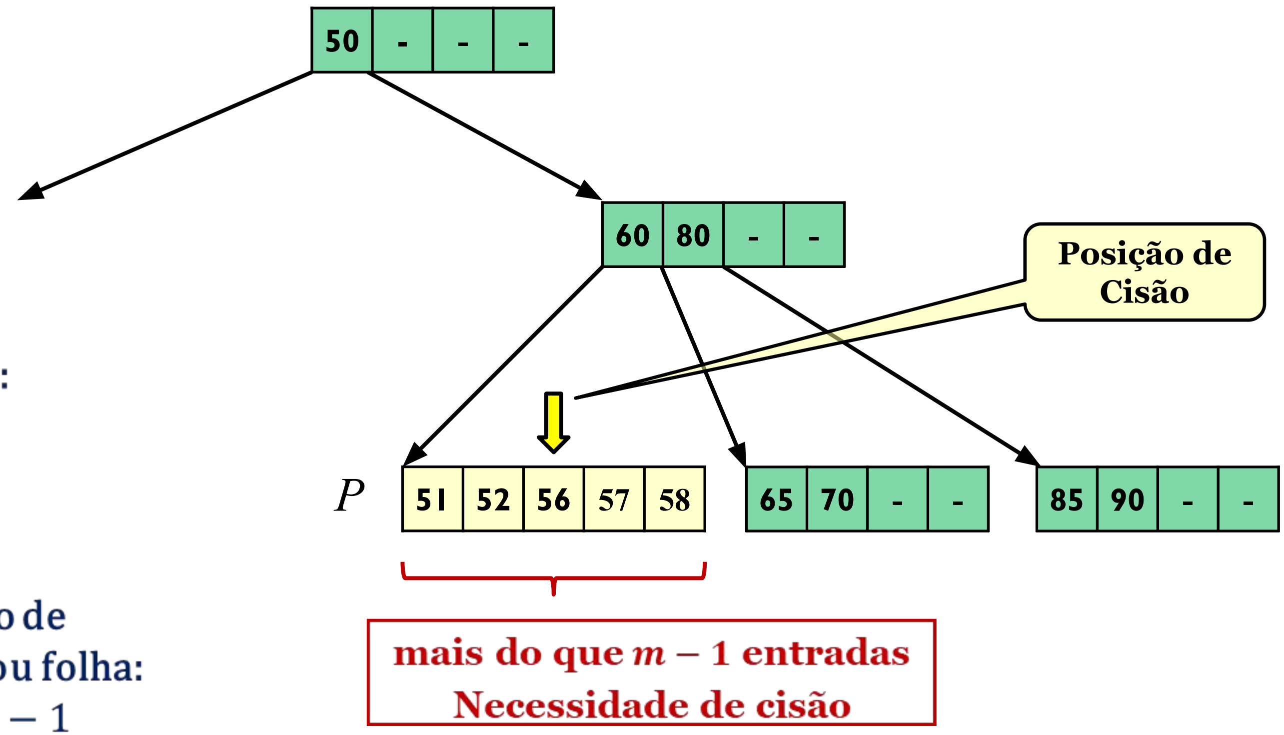
Inserção

- Ilustra-se a inserção da chave y=57 na árvore B com ordem m=5.



Árvore B
Inserção

- Ilustra-se a inserção da chave y=57 na árvore B com ordem m=5.

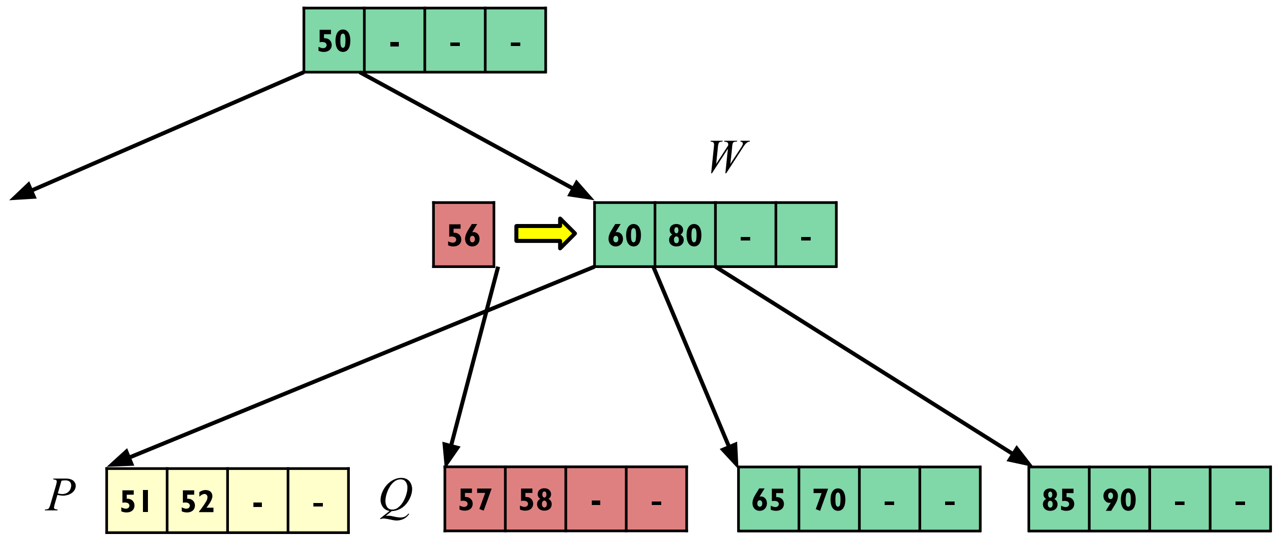


Variação do número de filhos de um nó interno:
$\lceil m/2 \rceil \leq k \leq m$
$3 \leq k \leq 5$

Variação do número de entradas de nó interno ou folha:
$\lceil m/2 \rceil - 1 \leq l \leq m - 1$
$2 \leq l \leq 4$

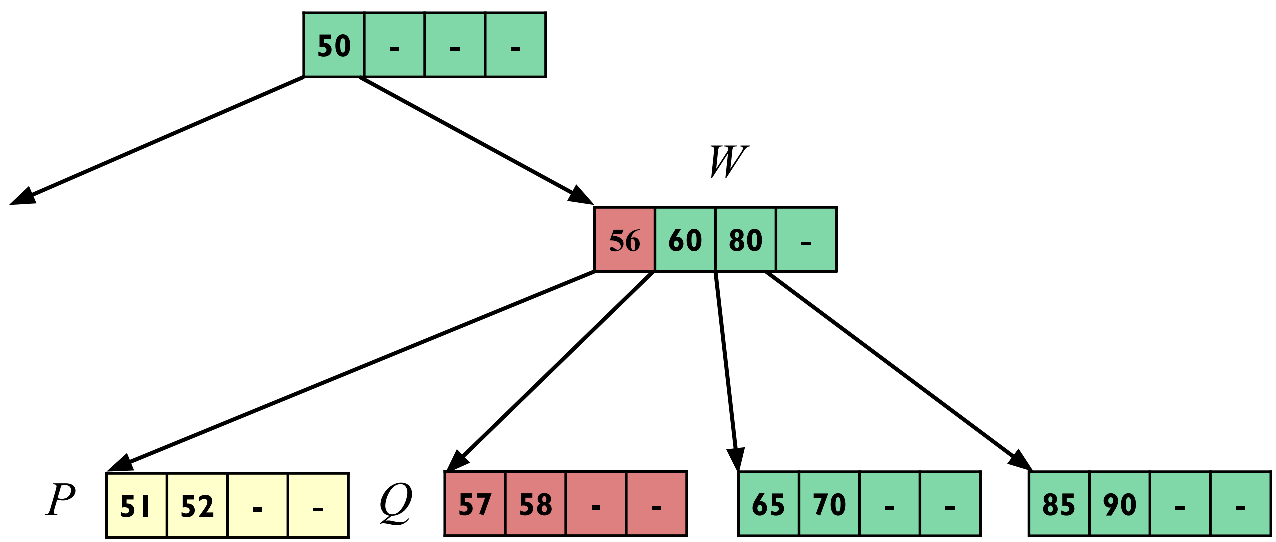
Árvore B
Inserção

- Ilustra-se a inserção da chave y=57 na árvore B com ordem m=5.



Árvore B
Inserção

- Ilustra-se a inserção da chave y=57 na árvore B com ordem m=5.

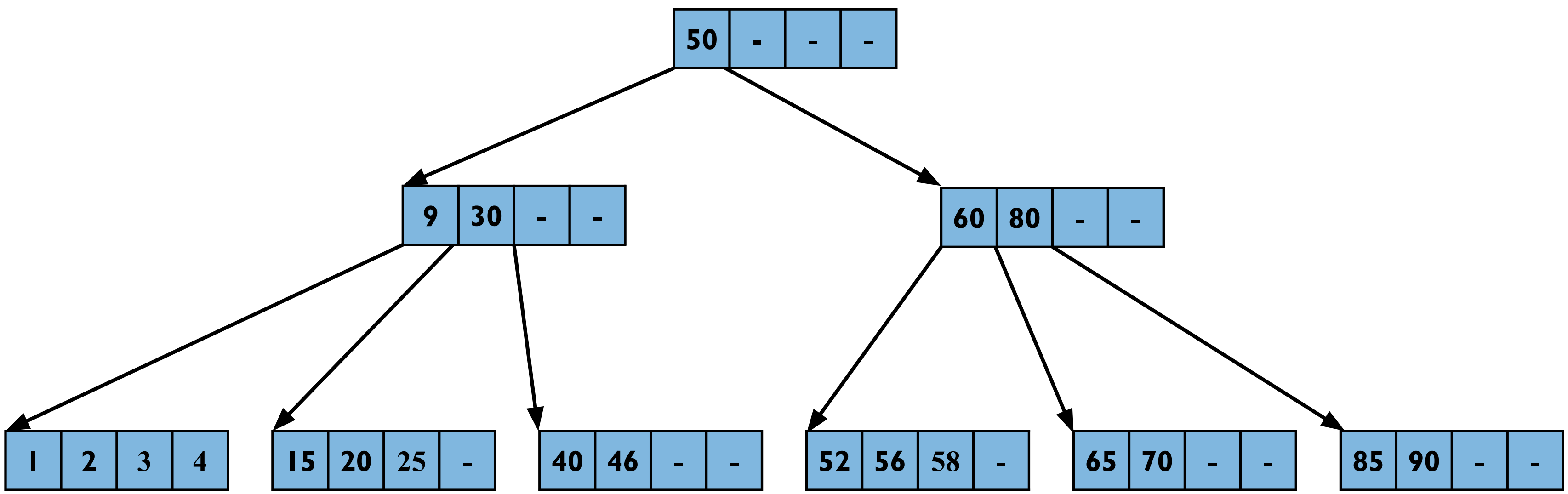


Árvore B
Remoção

- A remoção de um elemento em uma árvore B de ordem m, pode ser considerada como o processo inverso ao da inserção.
- O primeiro passo da remoção consiste na busca pelo nó que contém o elemento a ser removido. Chamaremos a este nó de nó origem P.
- Um problema que deve ser tratado ao se remover um novo elemento em um nó é a possibilidade de acontecer um ***underflow*** no nó origem da remoção. Isto significa que o nó origem P, possui o número mínimo de chaves necessárias $\lceil m/2 \rceil - 1$, após a remoção violará as propriedades de árvore B.
- Neste caso, remover um novo elemento produz *underflow* ($\lceil m/2 \rceil - 2$ chaves) o que deve ser tratado mediante a concatenação do nó origem P junto a o seu nó irmão Q.
- A concatenação de dois nós é propagável é pode atingir a raiz.

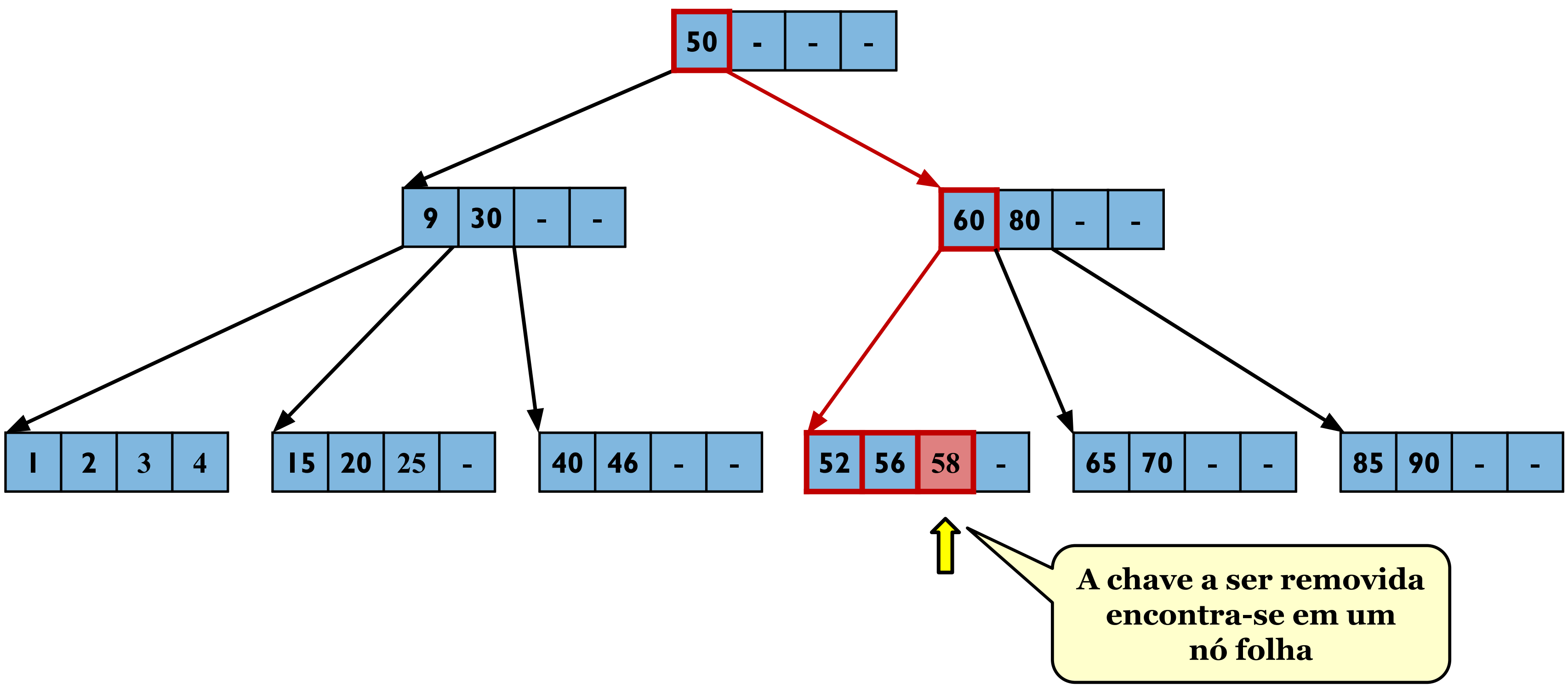
Árvore B
Remoção

- Ilustra-se a remoção da chave y=58 na árvore B com ordem m=5.



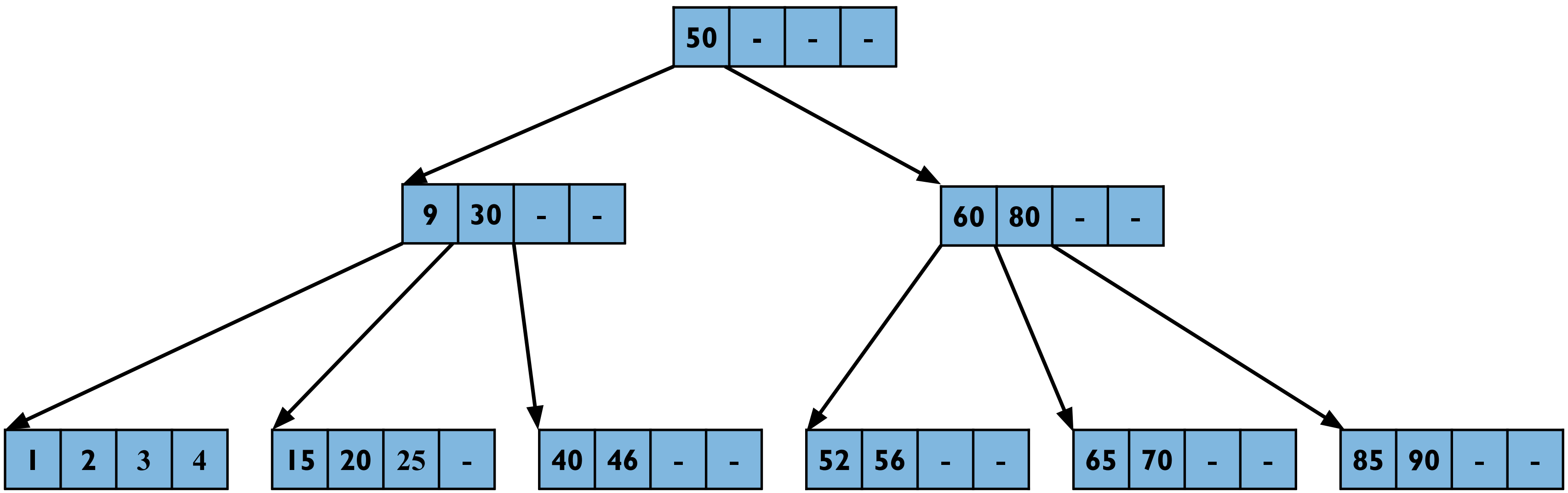
Árvore B
Remoção

- Ilustra-se a remoção da chave y=58 na árvore B com ordem m=5.



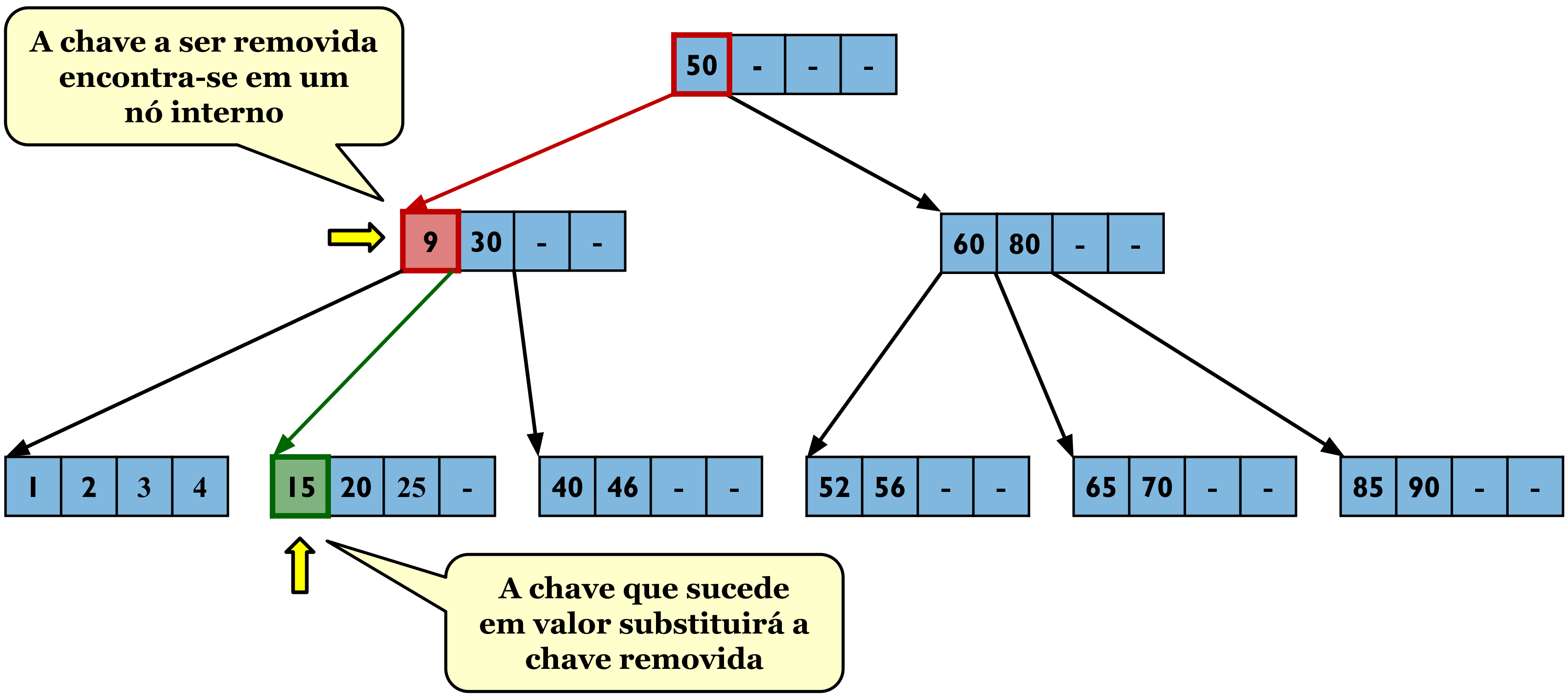
Árvore B
Remoção

- Ilustra-se a remoção da chave y=9 na árvore B com ordem m=5.



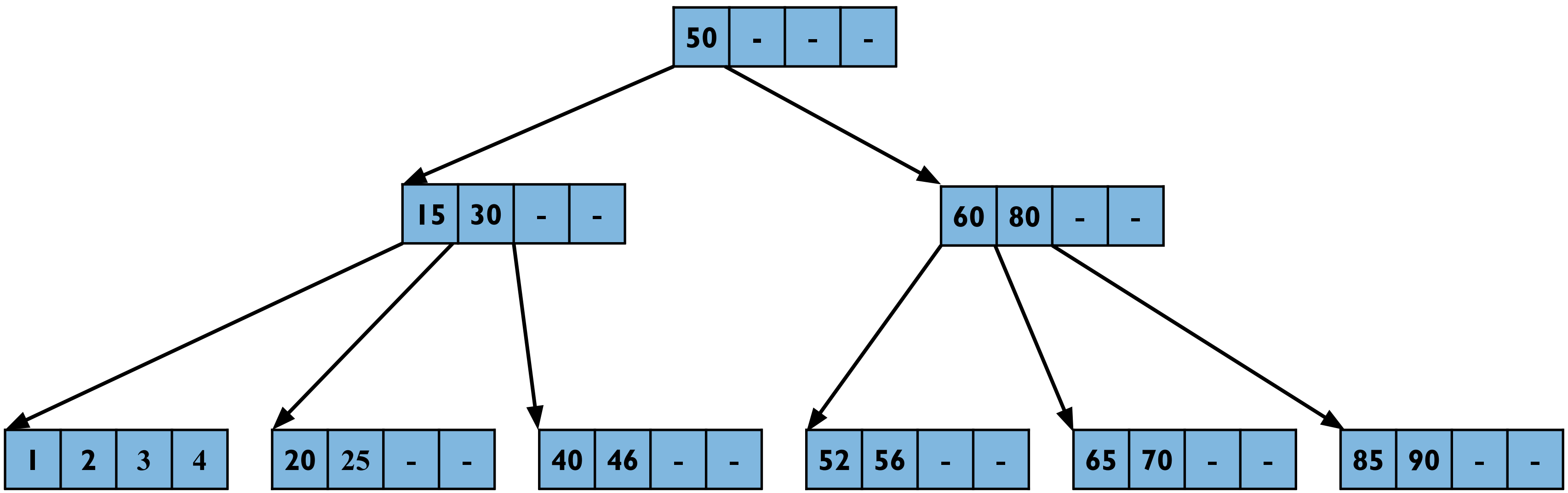
Árvore B
Remoção

- Ilustra-se a remoção da chave y=9 na árvore B com ordem m=5.



Árvore B
Remoção

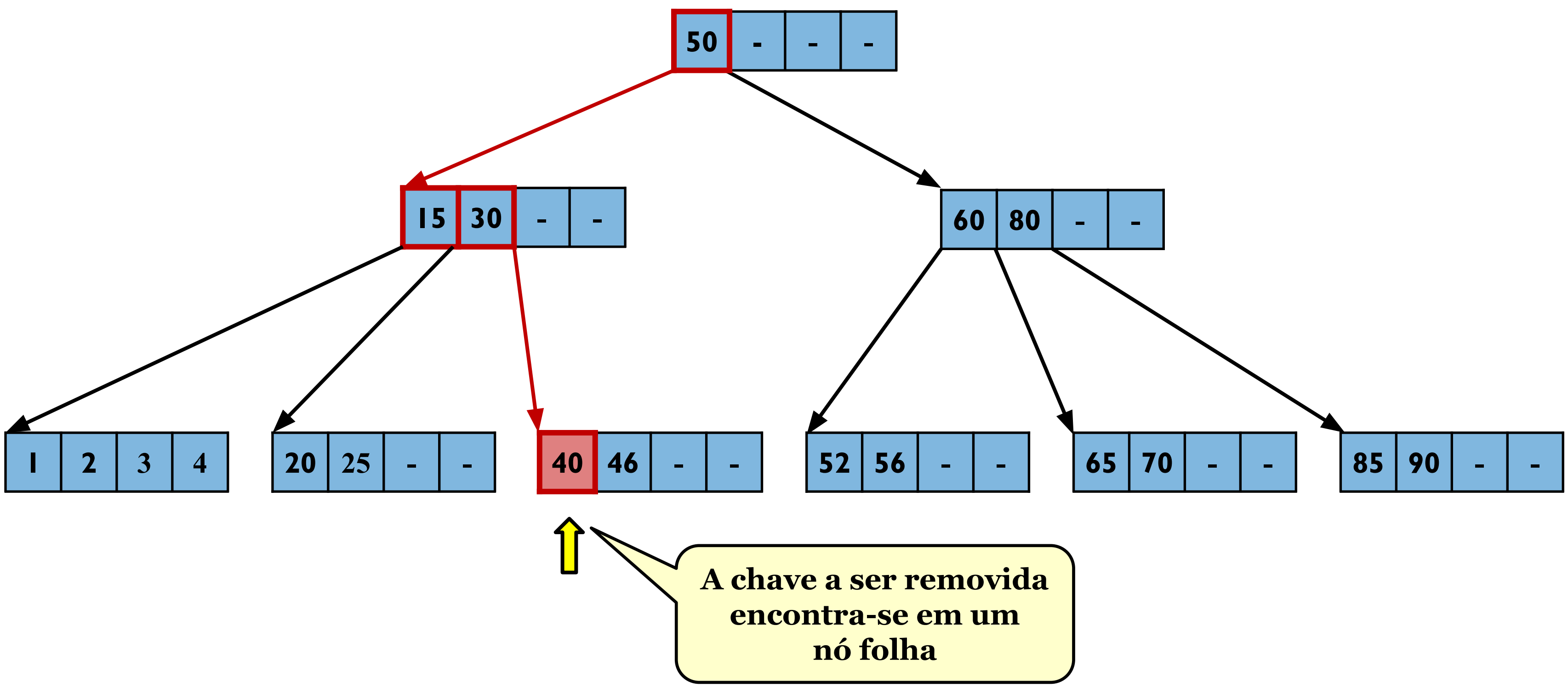
- Ilustra-se a remoção da chave y=40 na árvore B com ordem m=5.



Árvore B

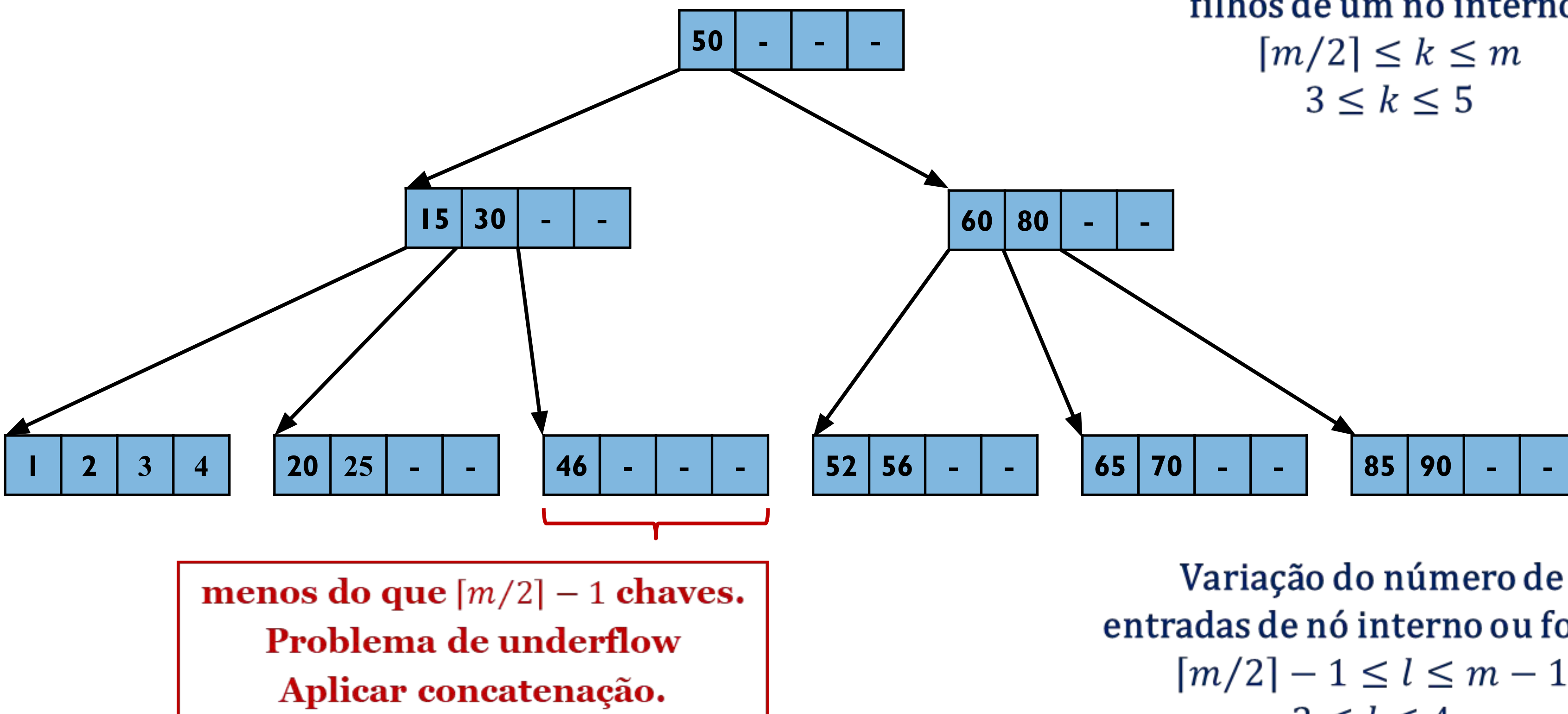
Remoção

- Ilustra-se a remoção da chave y=40 na árvore B com ordem m=5.



Árvore B
Remoção

- Ilustra-se a remoção da chave y=40 na árvore B com ordem m=5.



Variação do número de filhos de um nó interno:

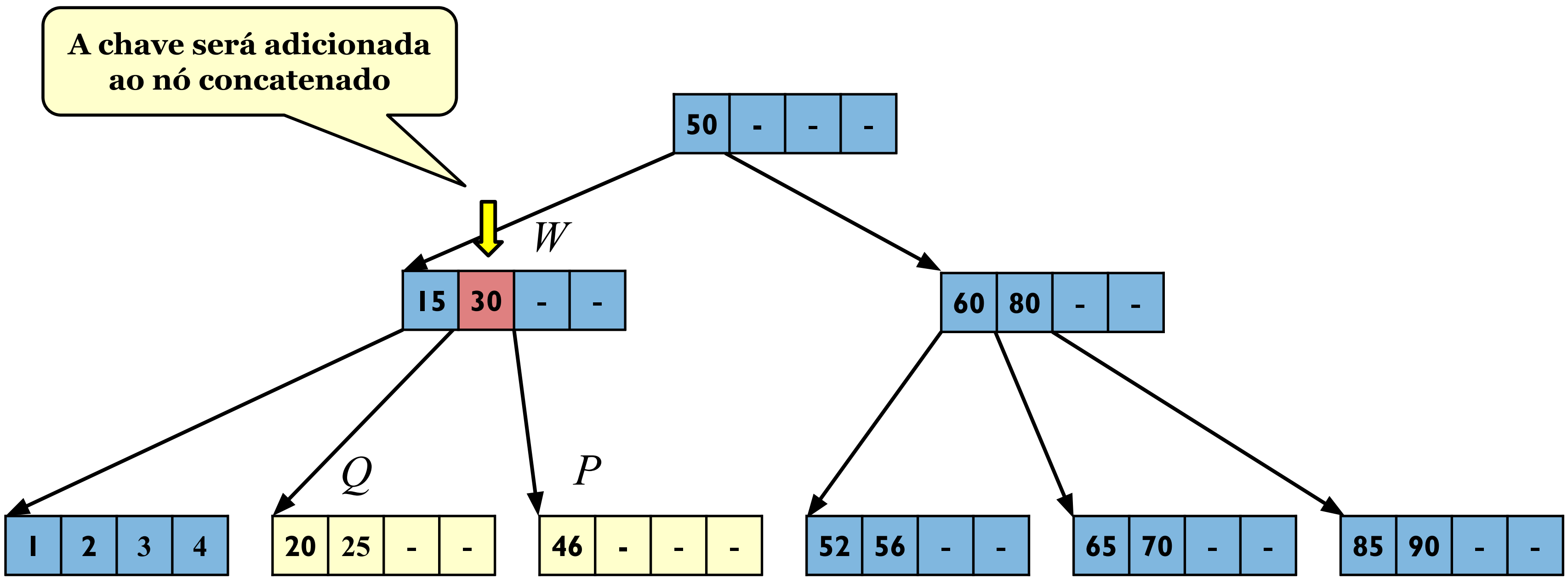
$$\lceil m/2 \rceil \leq k \leq m$$
$$3 \leq k \leq 5$$

Variação do número de entradas de nó interno ou folha:

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

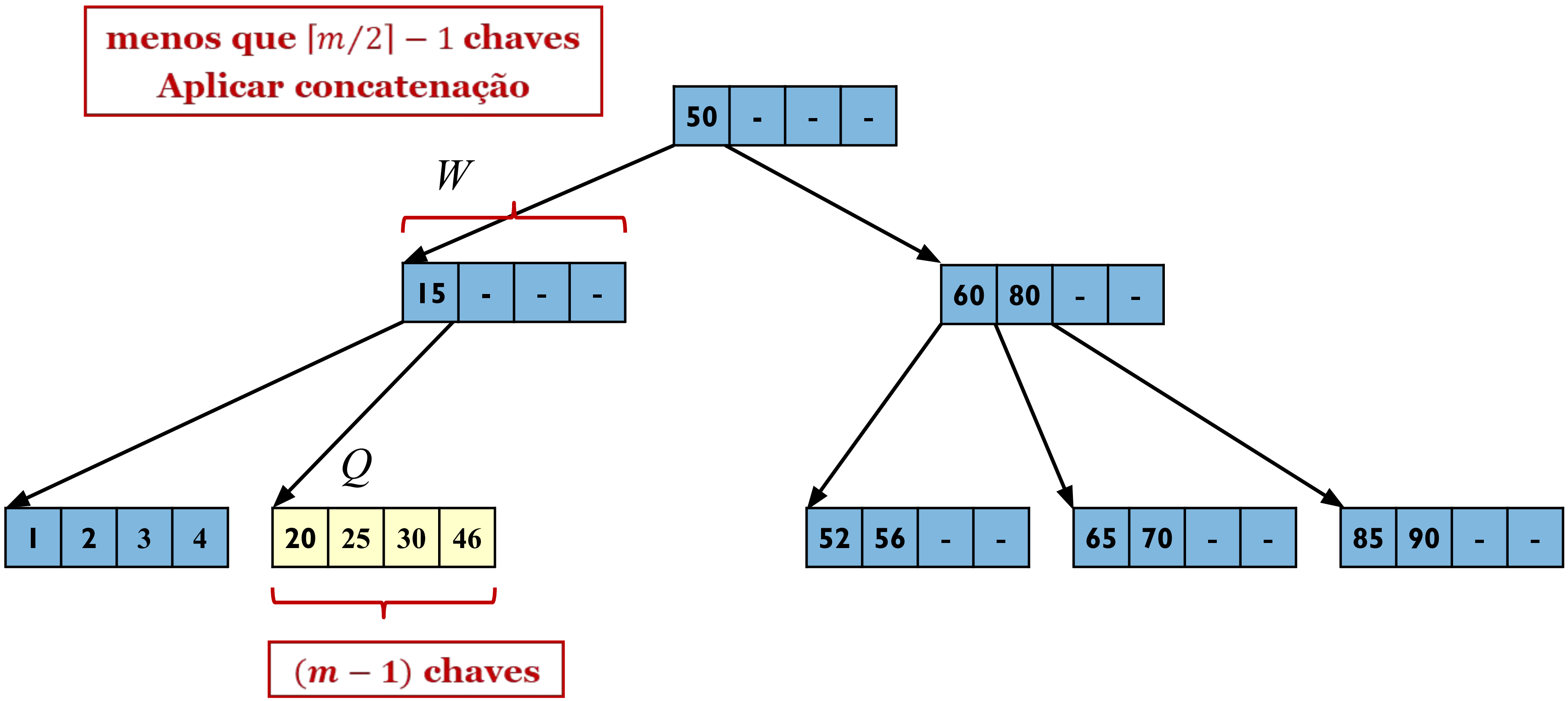
Árvore B
Remoção

- Ilustra-se a remoção da chave y=40 na árvore B com ordem m=5.



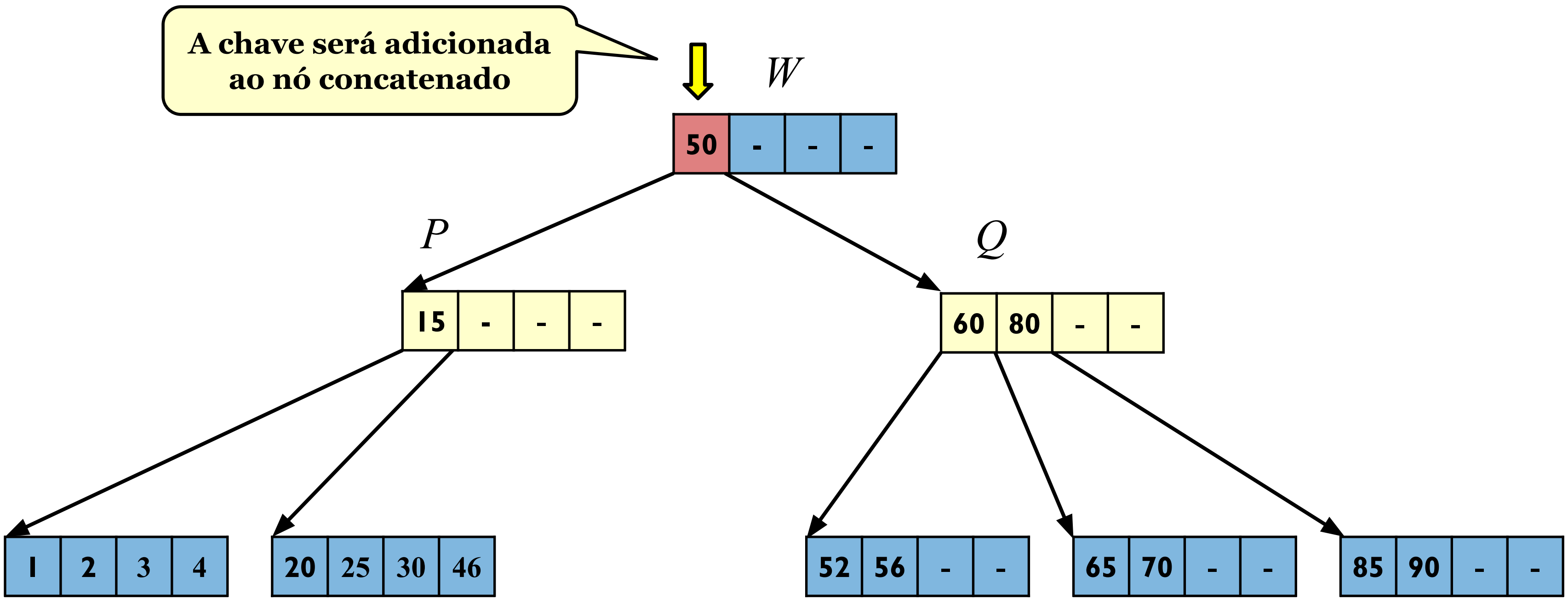
Árvore B
Remoção

- Ilustra-se a remoção da chave y=40 na árvore B com ordem m=5.



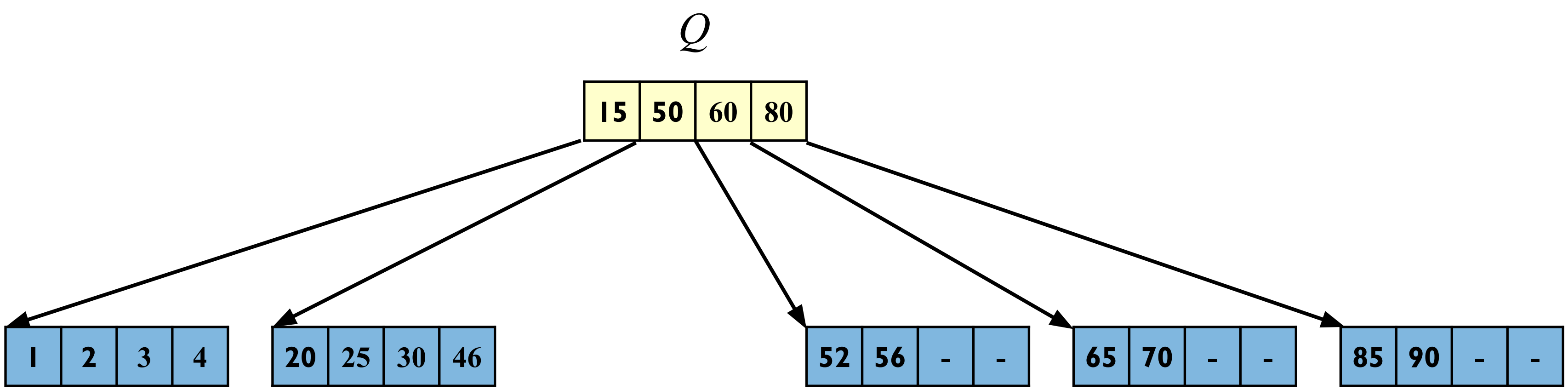
Árvore B
Remoção

- Ilustra-se a remoção da chave y=40 na árvore B com ordem m=5.



Árvore B
Remoção

- Ilustra-se a remoção da chave y=40 na árvore B com ordem m=5.



Árvores B*
Definição

- Uma árvore B* é uma variante da árvore B, introduzida por Donald Knuth e batizada por Douglas Comer.
- Em uma árvore B*, exige-se que todos os nós exceto a raíz, estejam pelo menos dois terços cheios, não apenas metade cheios, como nas árvores B.
- O número de chaves em todos os nós internos de uma árvore de ordem m é definido como k, onde $\left\lfloor \frac{2m-1}{3} \right\rfloor \leq k \leq m - 1$.
- Onde m é o número máximo de filhos de um nó interno.

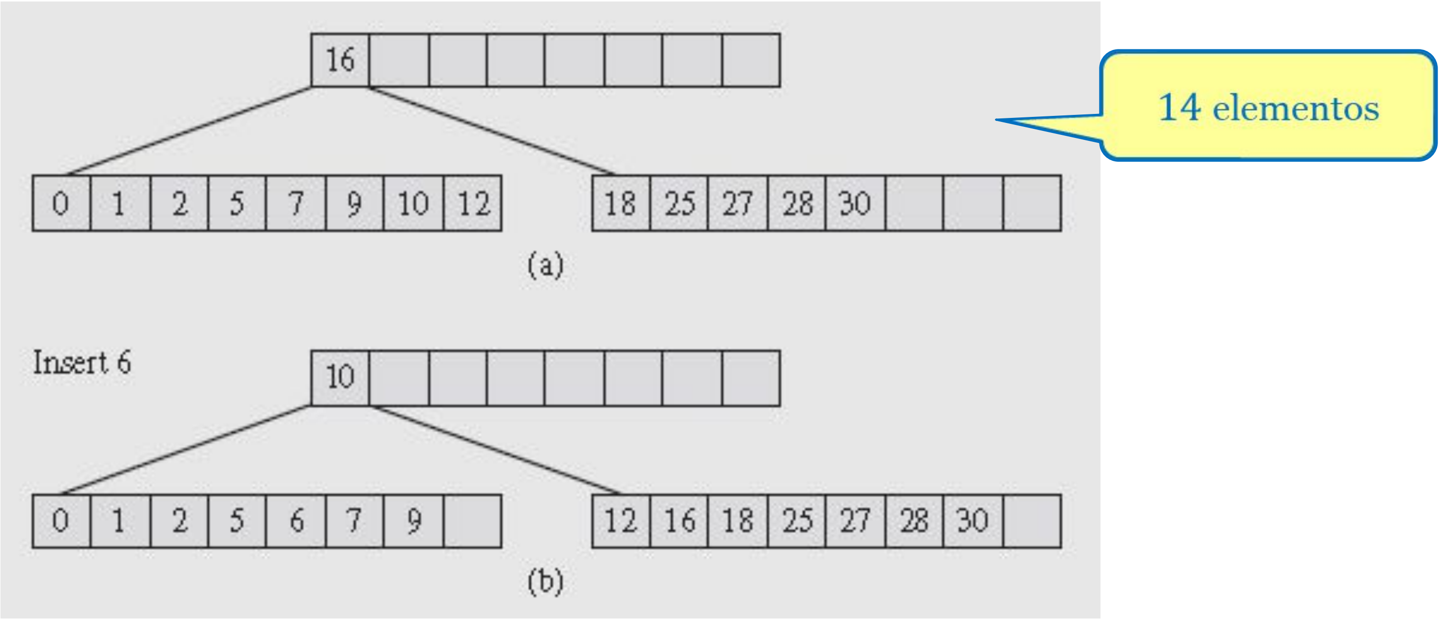
Árvores B*
Definição

- Com relação a árvore B, em uma árvore B* a frequência de divisão de nós é menor, atrasando-se o processo de divisão.
- A divisão também é diferenciada. Divide-se dois nós em três, e não um em dois como acontecia na árvore B.
- Uma divisão em uma árvore B* é atrasada tentando-se redistribuir as chaves entre um nó e seu irmão quando o nó transborda.

Árvores B*

Exemplo - Redistribuição

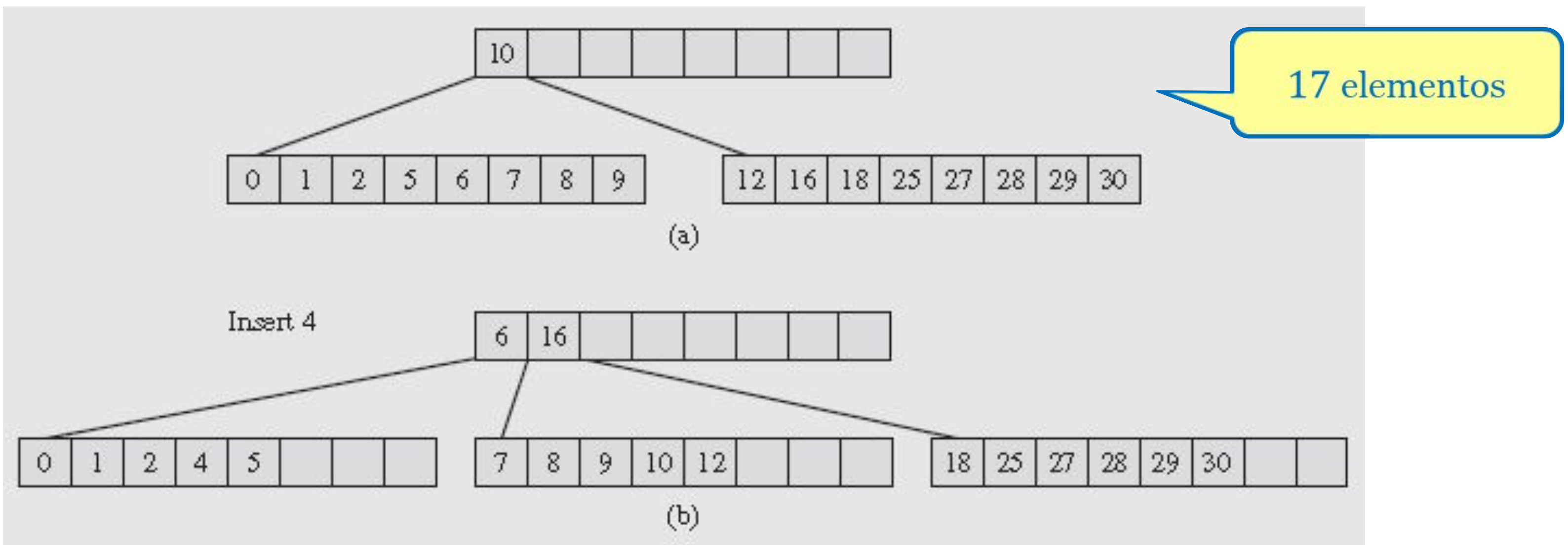
- A figura ilustra uma árvore B* de ordem 9. Procura-se inserir a chave 6. Essa chave deve ser inserida no nó esquerdo que já está cheio.
- Em vez de dividir o nó esquerdo, todas as chaves desse nó e de seu irmão são repartidas homogeneamente, considerando inclusive a chave intermediária no ascendente. A chave média, a chave 10 é colocada no ascendente.



- Observe que isto, distribui homogeneamente tanto as chaves como o espaço livre.

Árvores B*
Exemplo - Divisão

- A figura ilustra outro exemplo, uma árvore B* de ordem 9. Procura-se inserir a chave 4. Essa chave deve ser inserida no nó esquerdo que já está cheio. Além disso, o nó irmão também está cheio provocando uma divisão.



- Um novo nó é criado. As chaves do nó esquerdo e de seu irmão, junto com a chave de separação do nó ascendente são homogeneamente divididas entre três nós e duas chaves de separação são colocadas no nó ascendente (nó pai).
- Todos os três nós que participam da divisão são garantidos de estarem dois terços cheios.

Árvores B*
Comportamento

- As árvores B*, de acordo com a sua definição oferecem a garantia de estar sempre 66,6% cheias com relação ao espaço total alocado para elas, o que significa que no pior caso, 33,33% do espaço alocado pode ser desperdiçado.
- Na prática, estudos mostraram (Leung,1984) que a utilização média da árvore B* é de 81%.

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.