

Árvores

B



Árvore B

Uma **árvore B** é uma estrutura onde mais de uma chave é armazenada em cada nó, proporcionando uma organização de ponteiros tal que o tempo de acesso para buscas, inserções e remoções seja eficiente.

Sua construção assegura que todas as folhas estejam no mesmo nível, não importando a ordem de entrada dos dados.

Árvore B

Aplicações onde existe um número muito grande de dados e o armazenamento do conjunto de chaves não pode ser efetuado na memória principal.



Árvore B

Torna-se necessária a manutenção da tabela em memória secundária, aumentando significativamente o tempo de acesso de um nó.



A árvore B mantém mais de uma chave no mesmo nó. Possui uma organização de ponteiros que torna as operações de busca, inserção e remoção eficientes.

Árvore B

Considere d um número natural. Uma árvore B de ordem d é uma árvore que é vazia ou satisfaz as seguintes condições

(i) a raiz é uma folha ou tem, no mínimo, dois filhos.

(ii) cada nó diferente das folhas e da raiz tem, no mínimo, $d + 1$ filhos

(iii) cada nó tem no máximo $2d + 1$ filhos.

Árvore B

Um nó de uma árvore B é chamado de **PÁGINA** e apresenta as seguintes propriedades.

(i) cada página possui entre d e $2d$ chaves, exceto o nó raiz que possui entre 1 e $2d$ chaves.

(ii) seja m o número de chaves em uma página P . Então P possui $m + 1$ filhos.

(iii) em cada página P as chaves estão ordenadas: s_1, \dots, s_m , $d \leq m \leq 2d$, exceto a página raiz onde $1 \leq m \leq 2d$.

Árvore B

P possui $m + 1$ ponteiros p_0, \dots, p_m para os filhos de P . Nas folhas, os ponteiros indicam λ .

p_0	$s_1 \ p_1$	$s_2 \ p_2$	\dots	$s_m \ p_m$
-------	-------------	-------------	---------	-------------

(s_k, p_k) é uma entrada, $k > 0$

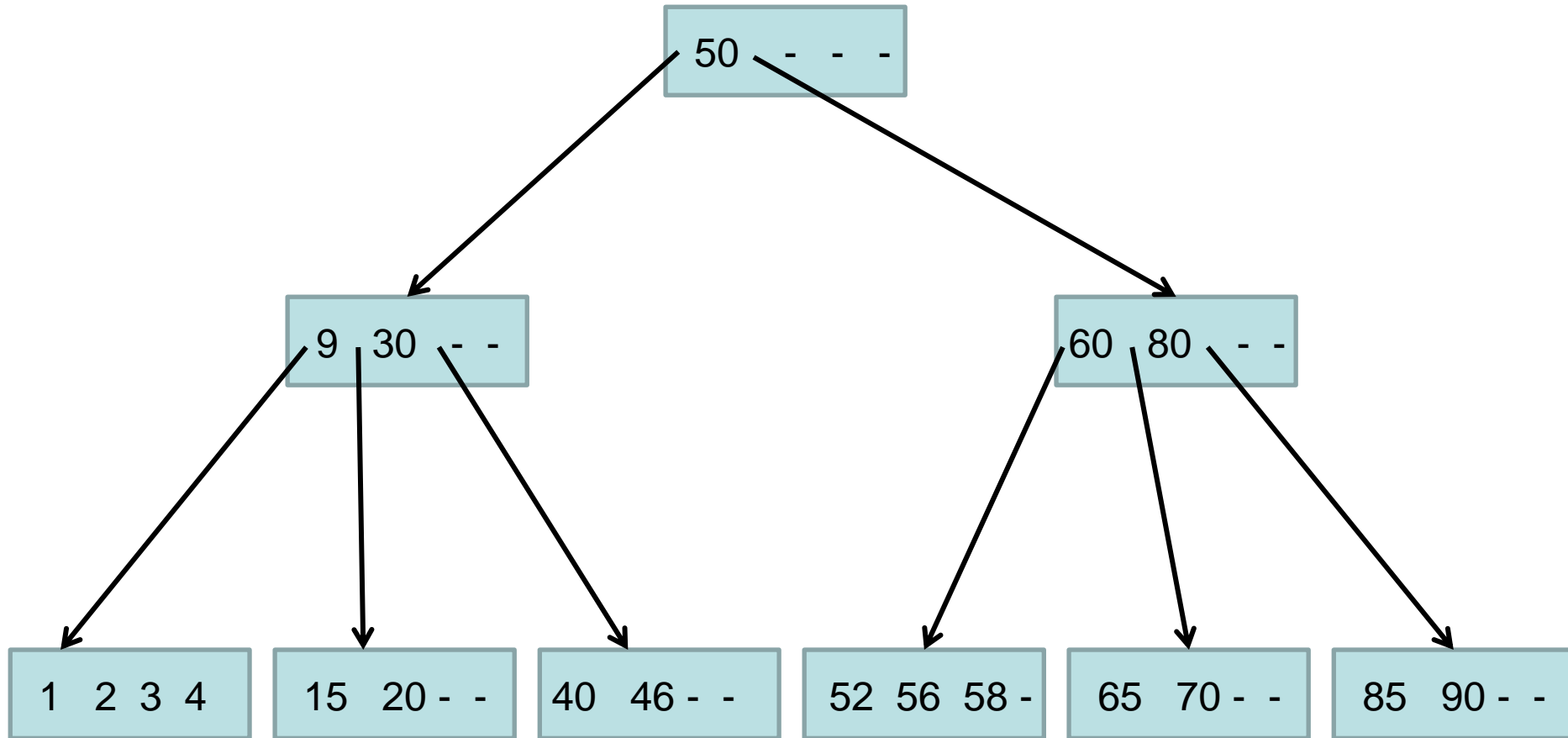
p_0 é a entrada, se $k = 0$

Árvore B

Considere P uma página com m chaves.

- Para qualquer chave y , pertencente à página apontada por p_0 , $y < s_1$;
 - Para qualquer chave y , pertencente à página apontada por p_k , $1 \leq k \leq m - 1$, $s_k < y < s_{k+1}$;
 - Para qualquer chave y , pertencente à página apontada por p_m , $y > s_m$.
-

Árvore B



Exercício

1. Desenhar uma árvore B de ordem 3 que contenha as seguintes chaves:

1, 3, 6, 8, 14, 32, 36, 38, 39, 41, 43.

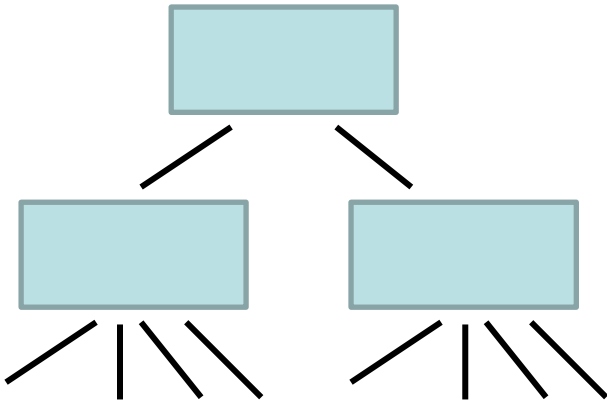
Árvore B

Número Mínimo e Máximo de Páginas em Árvore B

Seja uma árvore B de ordem d e altura h .

- A raiz possui, no mínimo, 2 filhos.
 - Cada página não folha \neq raiz possui, no mínimo, $d + 1$ filhos
-

Árvore B



Nível 1: 1 página

Nível 2: 2 páginas

Nível 3: $2(d+1)$ páginas

Nível i : $2(d+1)^{i-2}$ páginas

Nível h : $2(d+1)^{h-2}$ páginas

Árvore B

$$P_{\text{MIN}} = 1 + 2 [(d+1)^0 + (d+1)^1 + \dots + (d+1)^{h-2}]$$

$$P_{\text{MIN}} = 1 + 2/d [(d+1)^{h-1} - 1], \quad h \geq 1$$

$$P_{\text{MAX}} = \sum_{k=0}^{h-1} (2d+1)^k = 1/(2d) [(2d+1)^h - 1], \quad h \geq 1$$

Árvore B

Número Mínimo e Máximo de Chaves em Árvore B

$$n_{\text{MIN}} = 1 + d \left\{ \frac{2}{d} \left[(d+1)^{h-1} - 1 \right] \right\}$$

$$n_{\text{MIN}} = 2(d+1)^{h-1} - 1$$

$$n_{\text{MAX}} = 2d \left\{ \frac{1}{2d} \left[(2d + 1)^h - 1 \right] \right\}$$

$$n_{\text{MAX}} = (2d + 1)^h - 1$$

Árvore B

Altura h de uma Árvore B

$$\log_{2d+1}(n+1) \leq h \leq \log_{d+1}\left(\frac{n+1}{2}\right)$$

Para $n \geq 1$

Logo, a árvore B é balanceada.

Busca em Árvore B

Procedimento buscaB (x , pt , f , g)

// $f = 1$, sucesso; $f = 0$, fracasso

// g – posição do nó apontado por pt

$pt2 \leftarrow ptraiz$; $pt \leftarrow \lambda$; $f \leftarrow 0$

Enquanto ($pt2 \neq \lambda$) faça

$i \leftarrow g \leftarrow 1$; $pt \leftarrow pt2$

 Enquanto ($i \leq pt2.m$) faça

 se ($x > pt2.s[i]$) então $i \leftarrow g \leftarrow i+1$

 senão

 se ($x = pt2.s[i]$) então $pt2 \leftarrow \lambda$; $f \leftarrow 1$

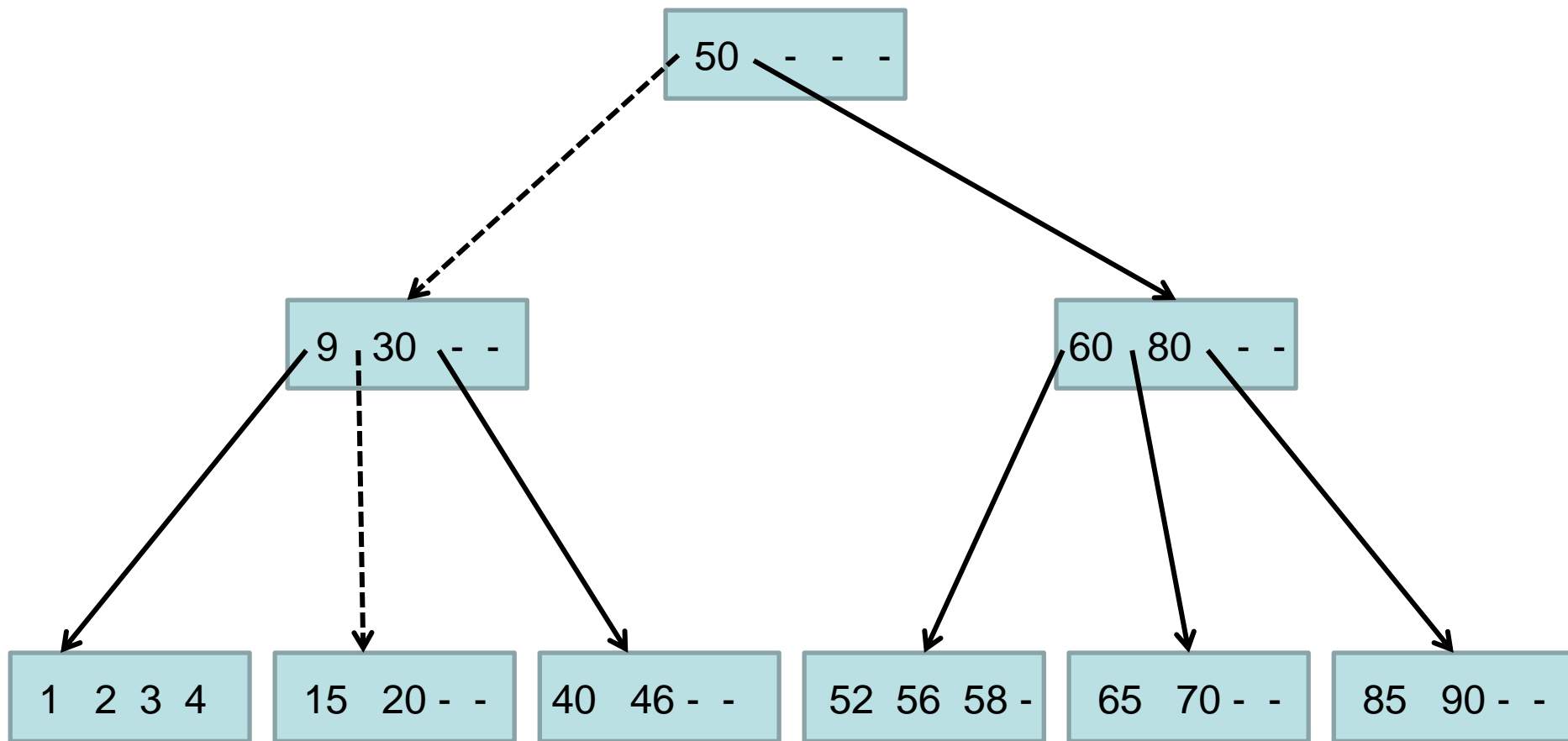
 senão $pt2 \leftarrow pt2.p[i-1]$

$i \leftarrow pt2.m+2$

 se ($i = pt2.m+1$) então $pt2 \leftarrow pt2.p[pt2.m]$

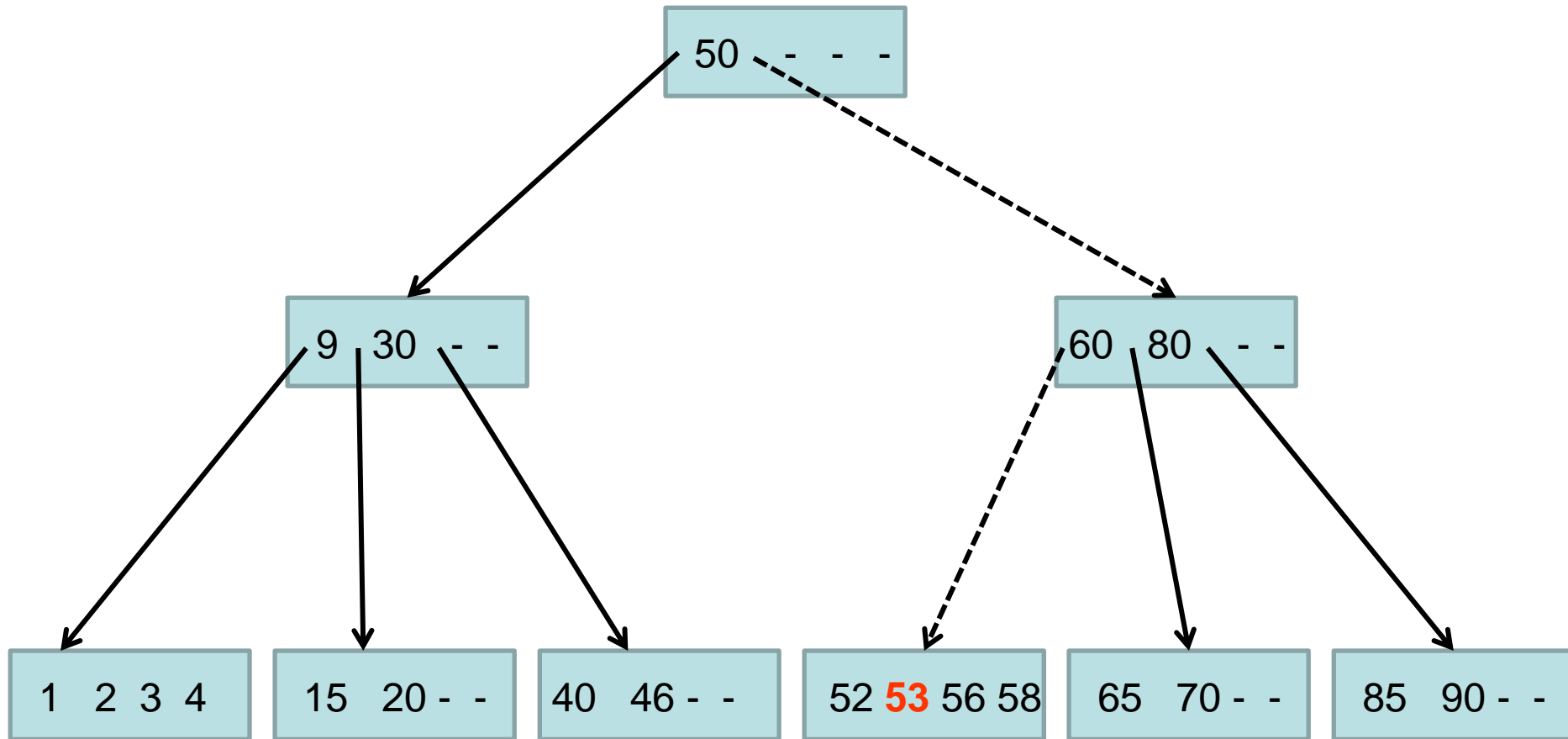
Busca em Árvore B

Exemplo: Busca da chave 20 (sucesso), chave 25 (fracasso)



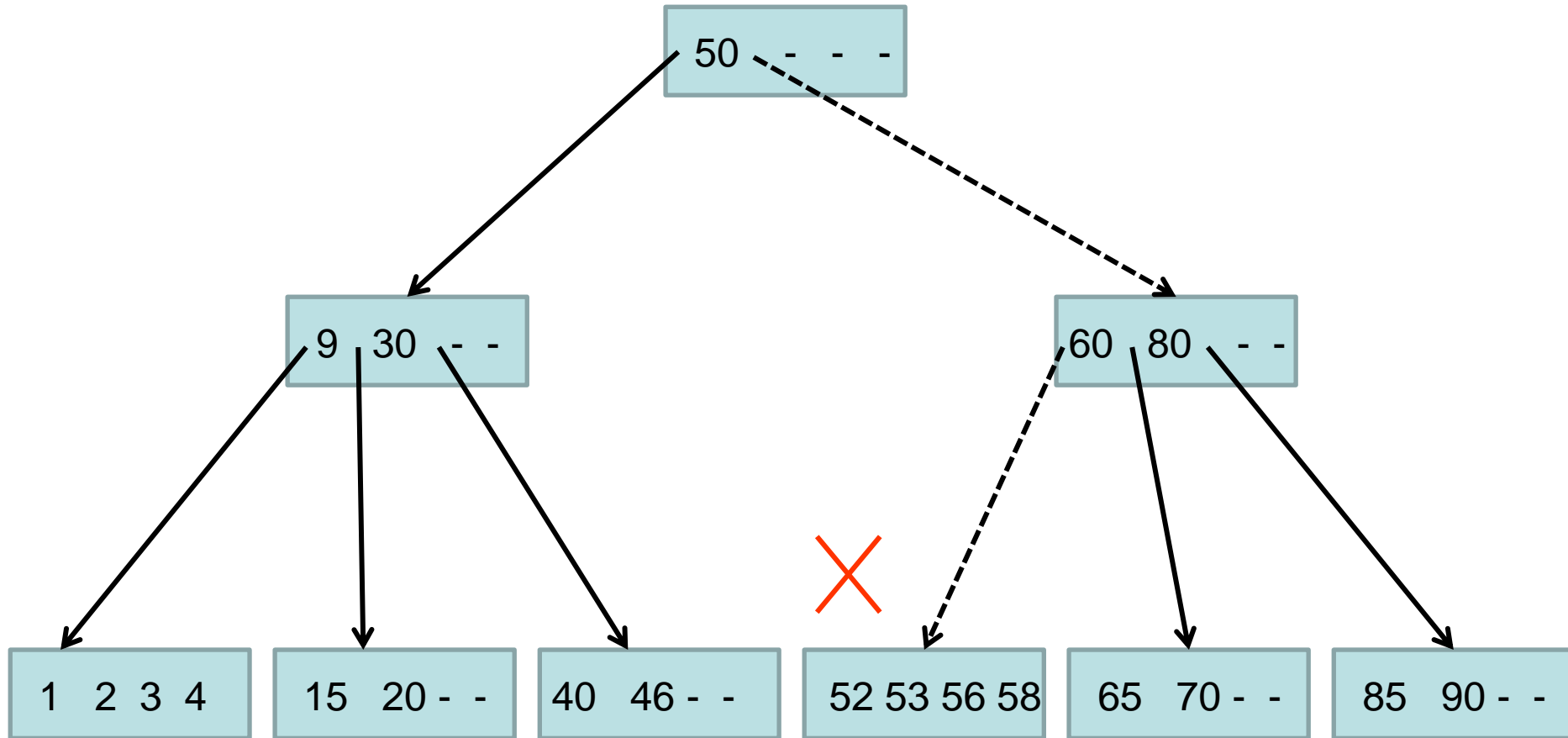
Inserção em Árvore B

Inserção da chave 53



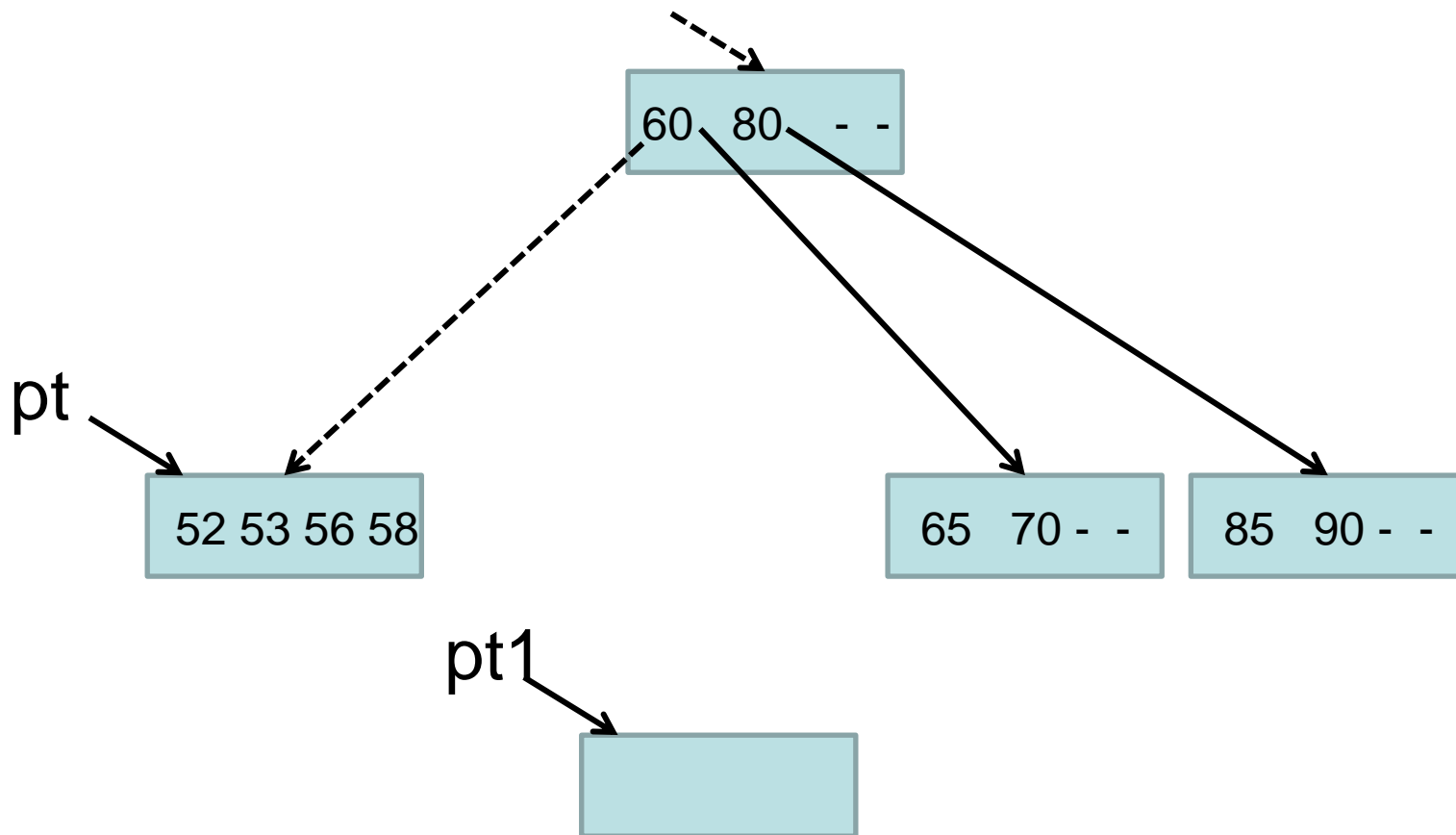
Inserção em Árvore B

Inserção da chave 59



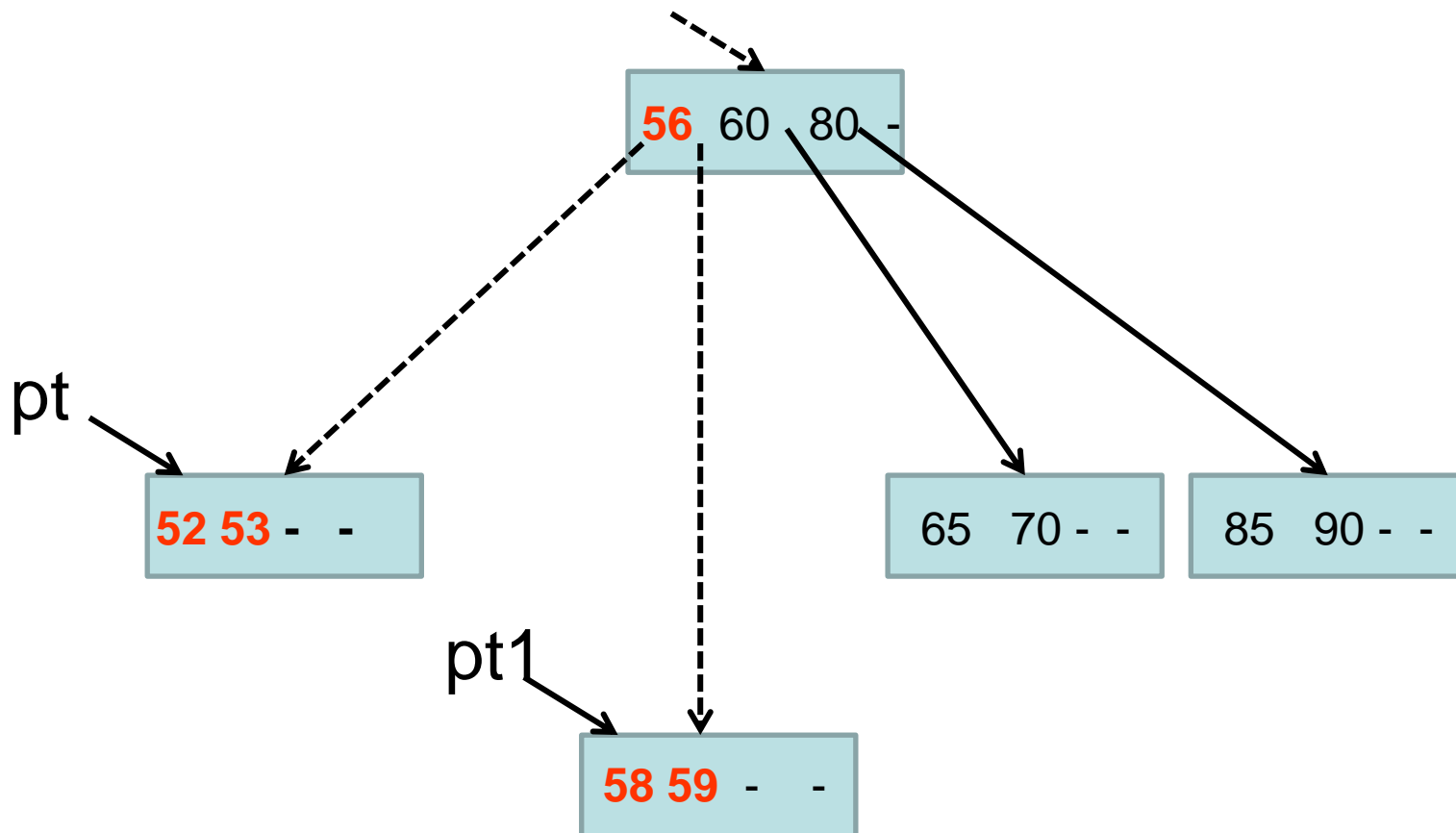
Inserção em Árvore B

Inserção da chave 59 – SOLUÇÃO: Cisão de Página



Inserção em Árvore B

Inserção da chave 59 – SOLUÇÃO: Cisão de Página



Inserção em Árvore B

Caso em que a página fica com $2d + 1$ chaves

Solução: **Cisão de Páginas**

Considere P a página com excesso de chaves. P será transformada em 2 páginas.

Considere pt o ponteiro que aponta para P.

Permanecerão **d chaves** em P. Uma nova página Q (apontada por pt1) é solicitada. Q recebe **d chaves**.

Inserção em Árvore B

A chave que sobra, da posição central de P , $s_d + 1$, forma com o ponteiro $pt1$, uma nova entrada na página W pai de P .

Final:

$P - p_0, (s_1, p_1), \dots, (s_d, p_d)$

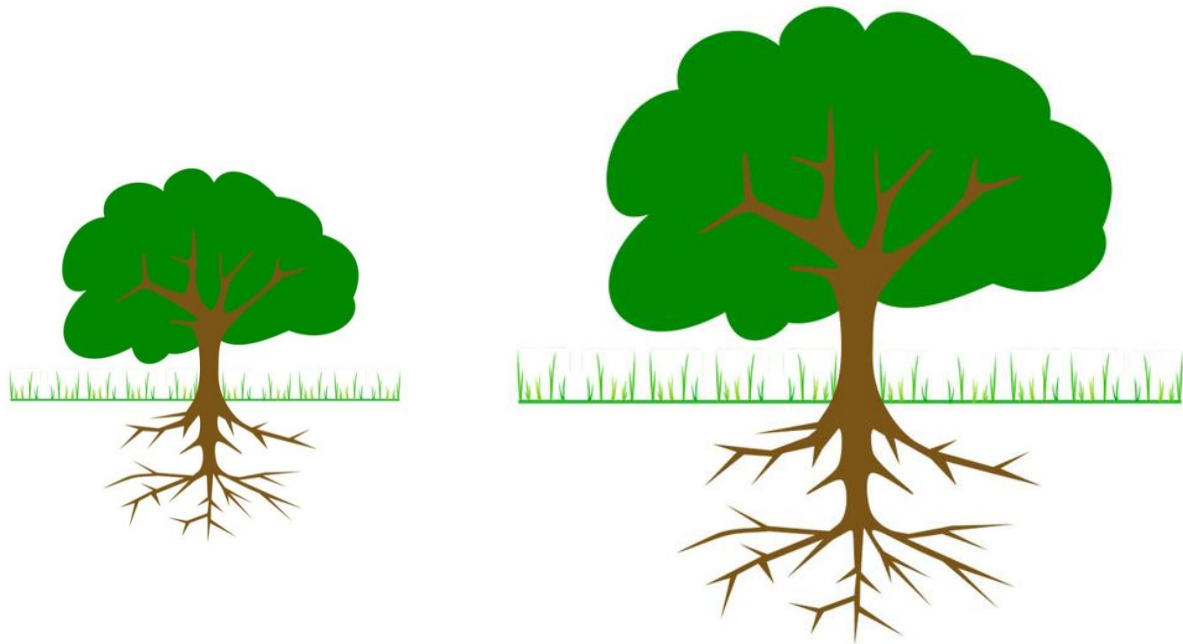
$Q - p_{d+1}, (s_{d+2}, p_{d+2}), \dots, (s_{2d+1}, p_{2d+1})$

$W - \text{recebe } (s_{d+1}, p_{d+1})$

OBS. Pode ser necessário fazer a cisão de W .

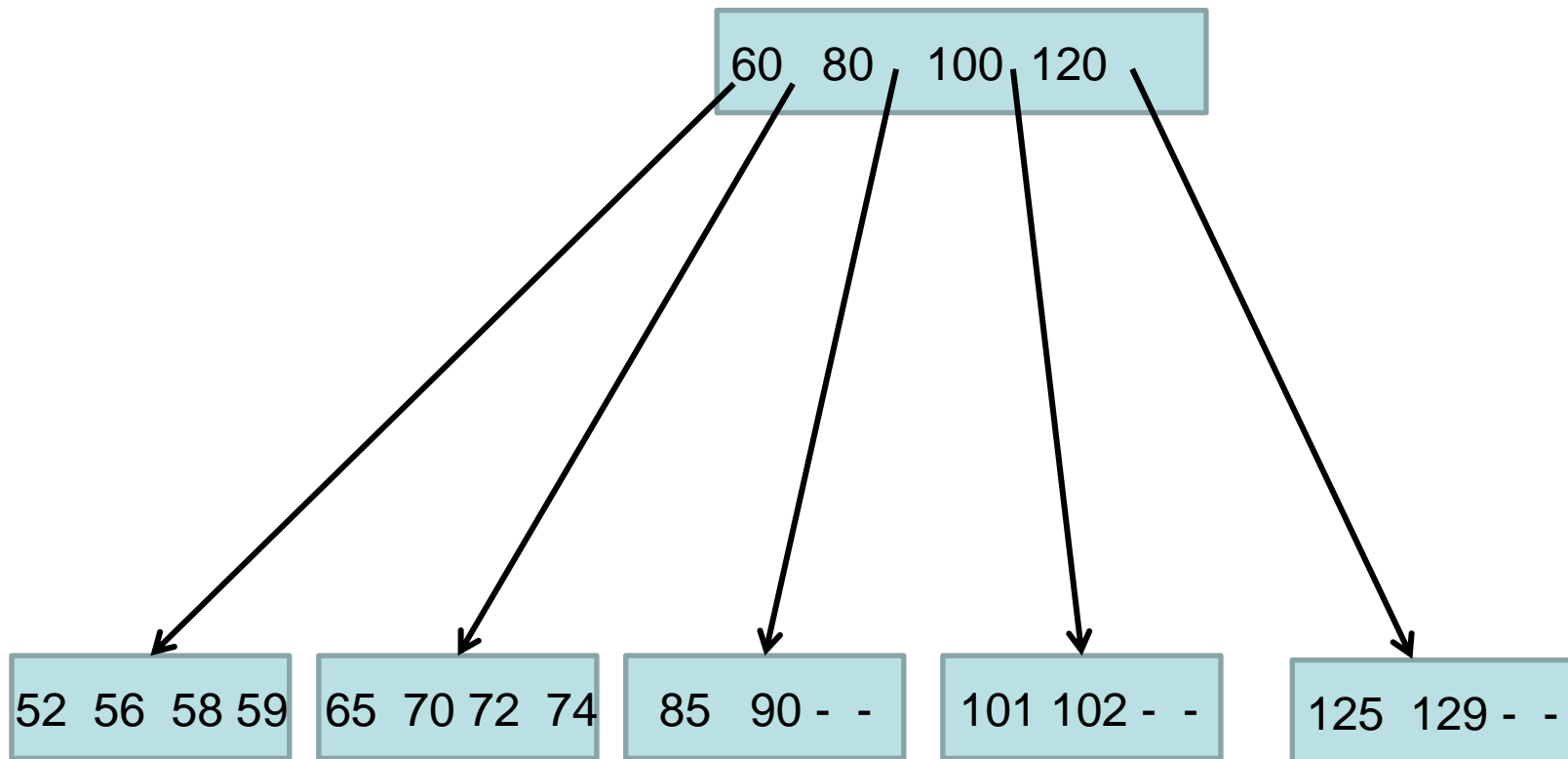
Inserção em Árvore B

A cisão é propagável e pode chegar a raiz da árvore, podendo acarretar o aumento da altura da árvore, que se mantem, entretanto, com as folhas no mesmo nível, uma vez que surge uma nova raiz.



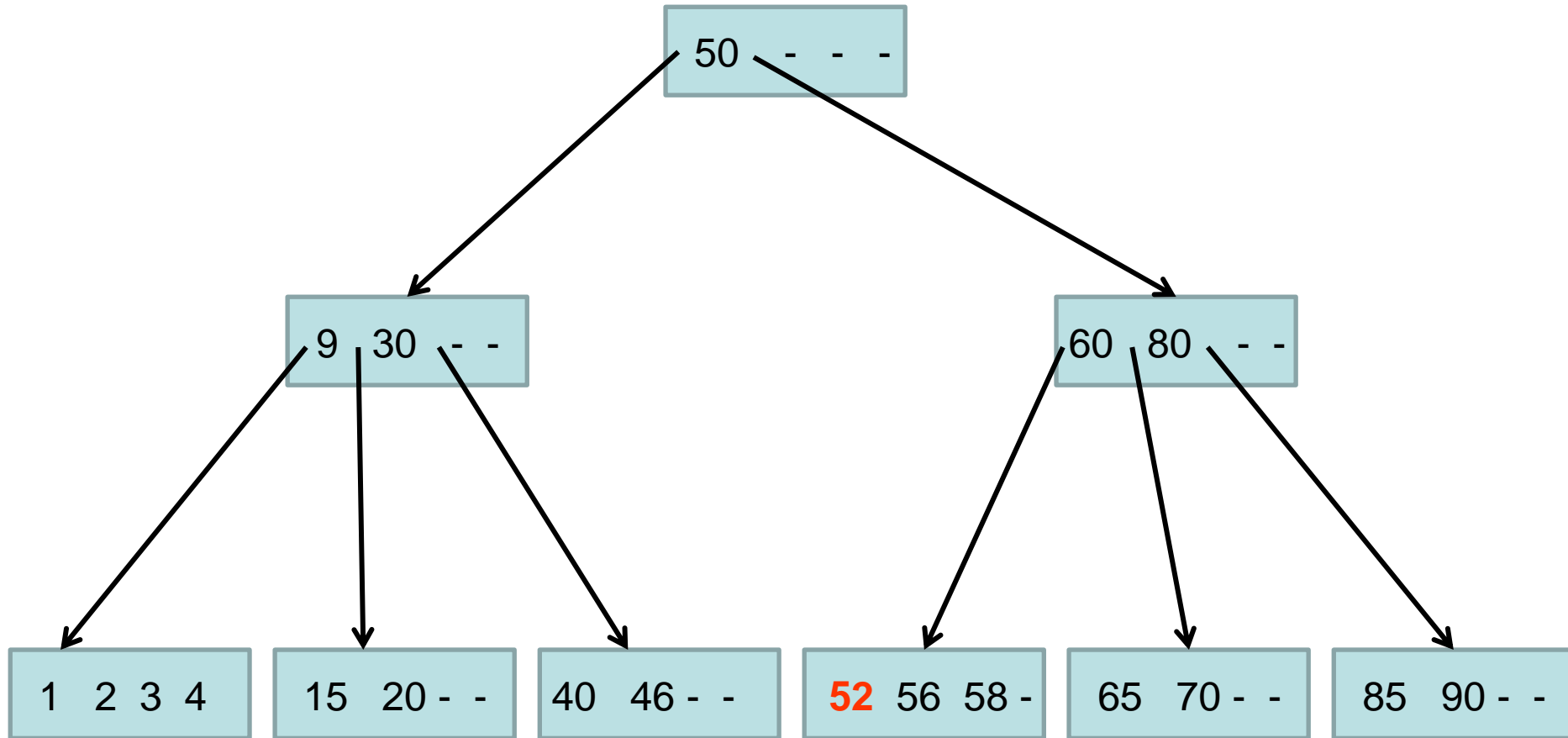
Exercício

1. Considerando a árvore B de ordem 2 abaixo, inserir a chave 73.



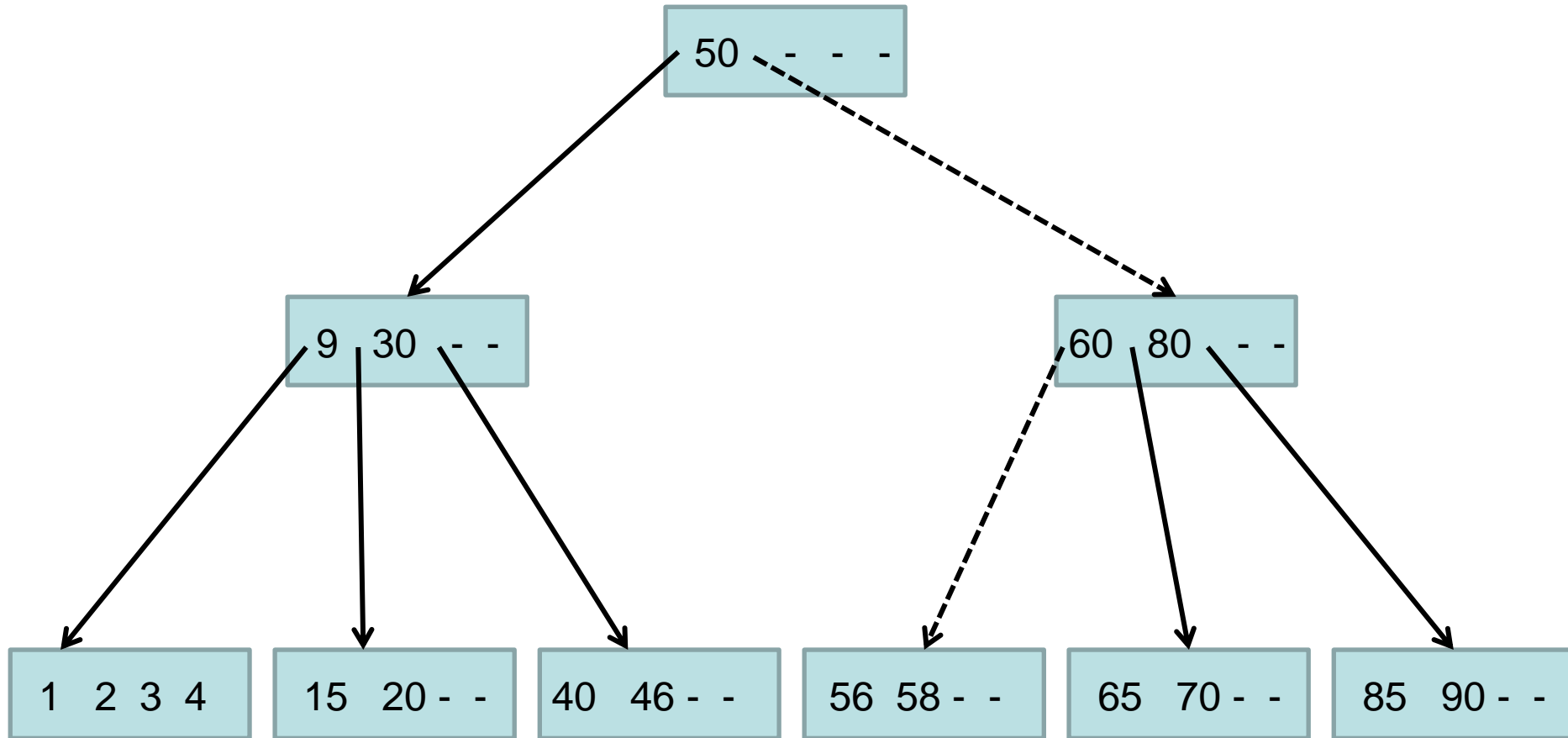
Remoção em Árvore B

Exemplo: Remoção da chave 52 (chave na folha)



Remoção em Árvore B

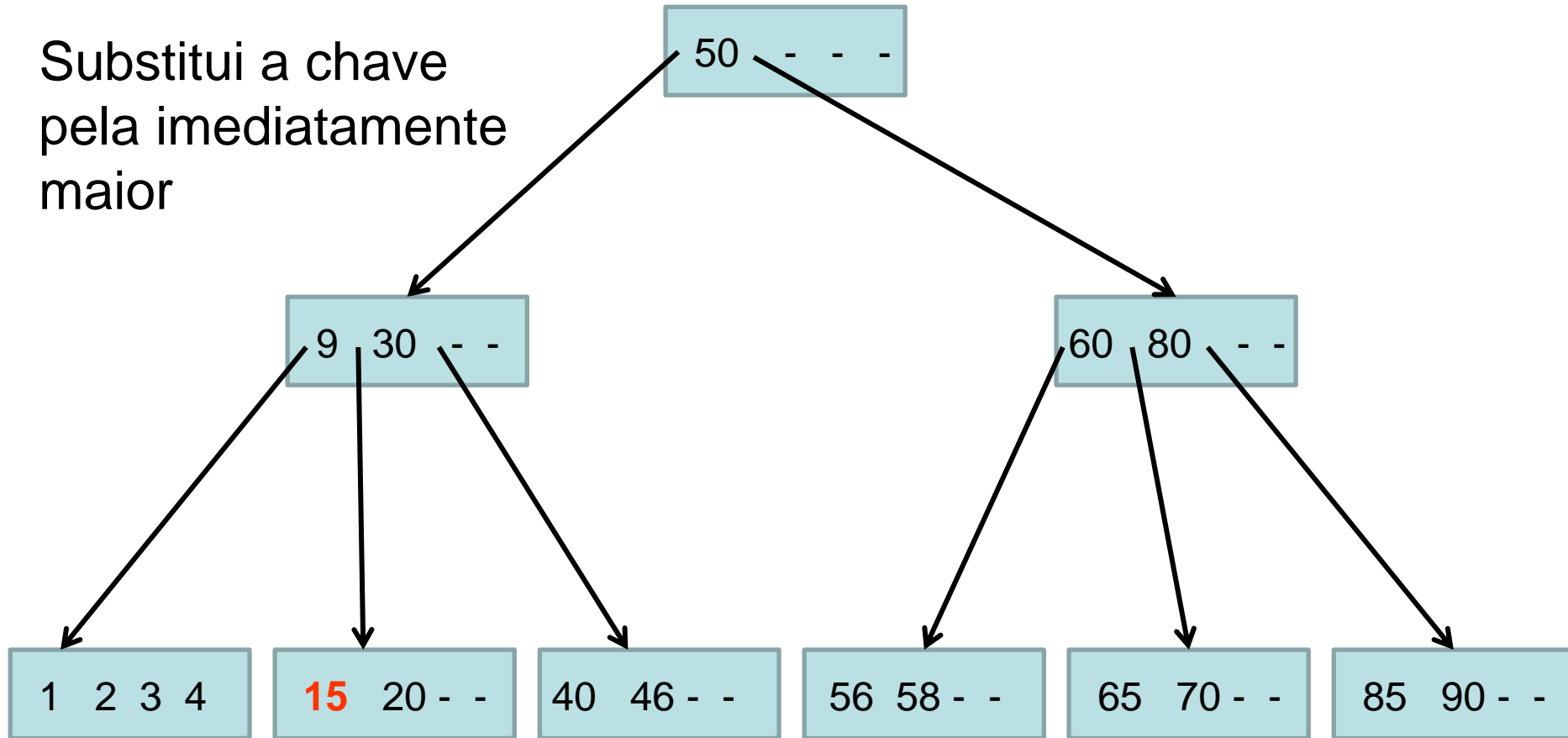
Exemplo: Remoção da chave 52 (chave na folha)



Remoção em Árvore B

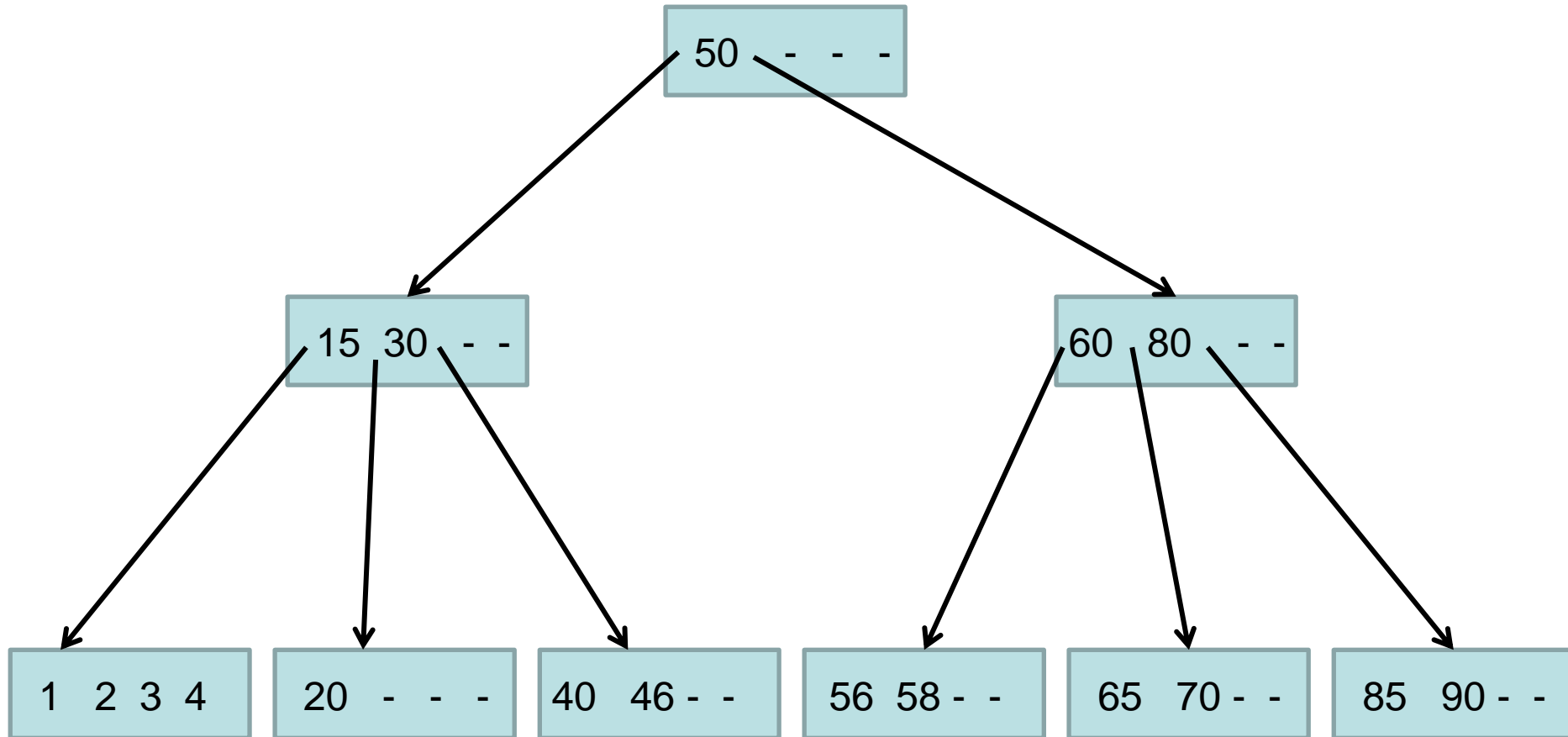
Exemplo: Remoção da chave 9 (nó não folha)

Substitui a chave
pela imediatamente
maior



Remoção em Árvore B

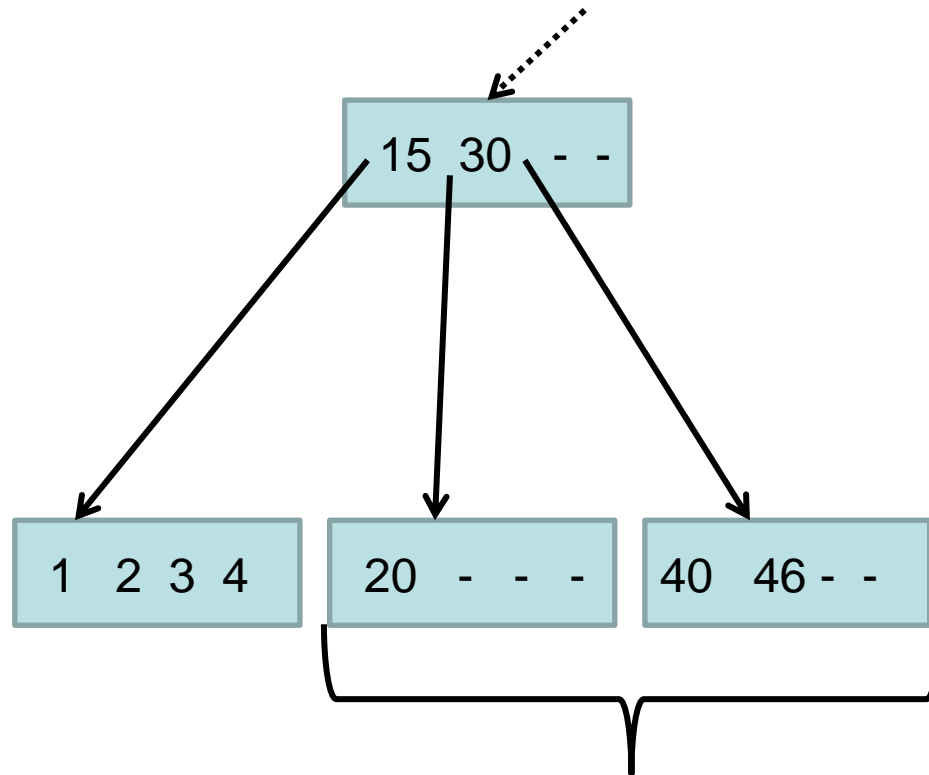
Exemplo: Remoção da chave 9 (nó não folha)



✗ Menos de d chaves !

Remoção em Árvore B

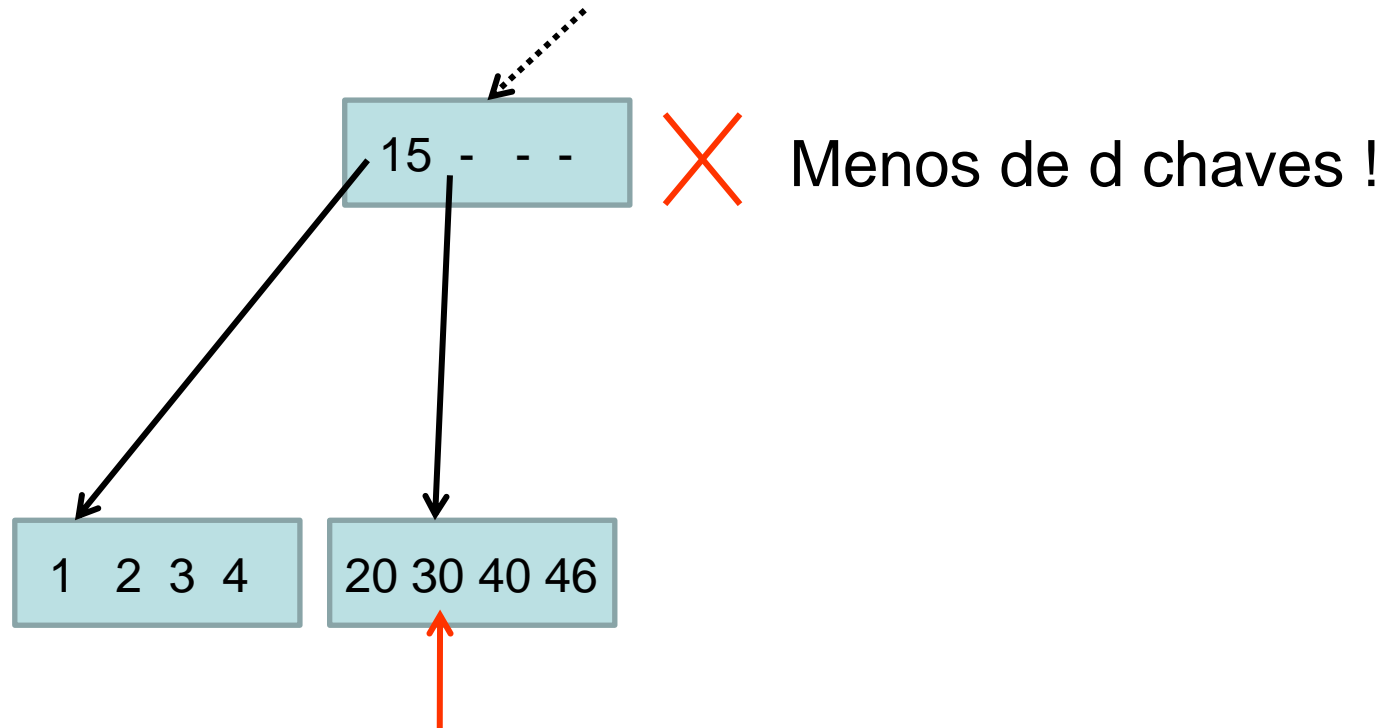
Solução: CONCATENAÇÃO



páginas irmãs que possuem juntas menos de $2d$ chaves

Remoção em Árvore B

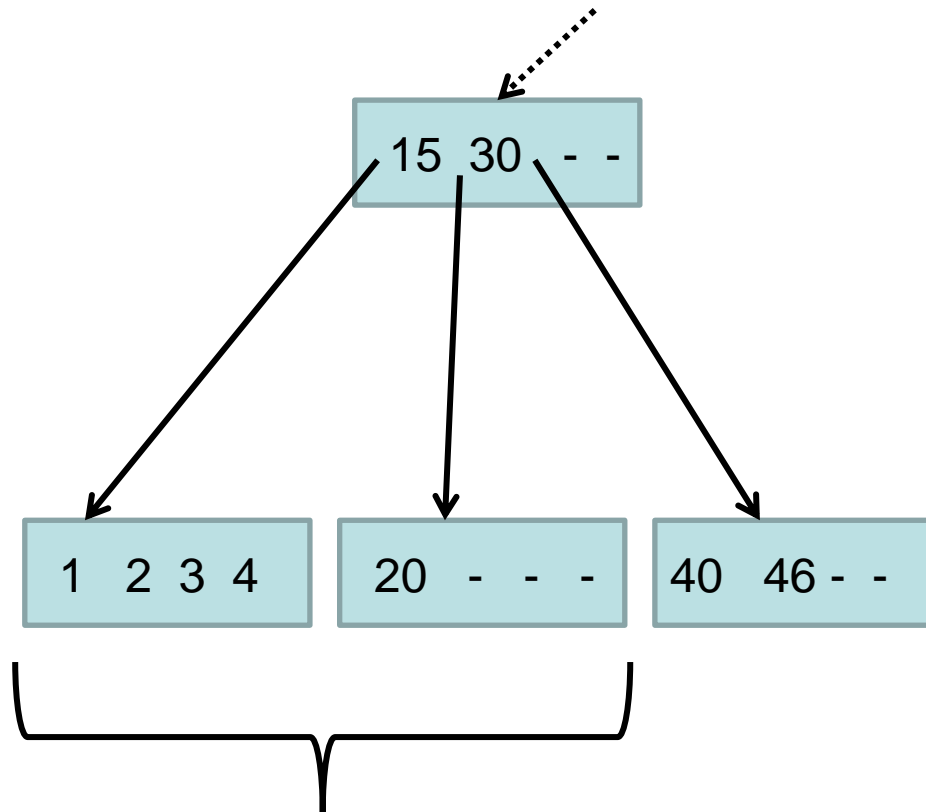
Solução: CONCATENAÇÃO



Entrada do pai que passa a fazer parte da chave resultante da concatenação

Remoção em Árvore B

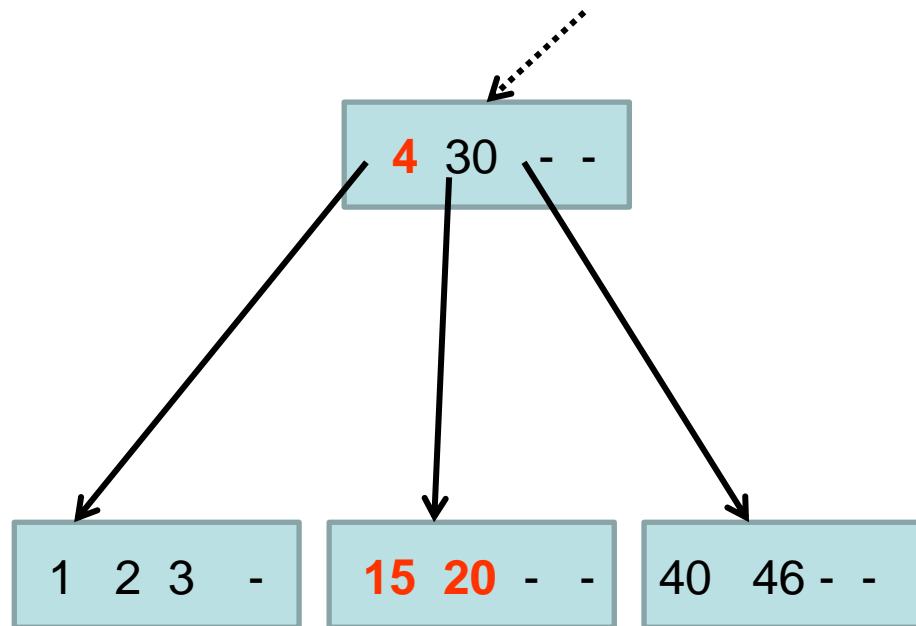
Solução: REDISTRIBUIÇÃO



páginas irmãs que possuem juntas mais de 2d chaves

Remoção em Árvore B

Solução: REDISTRIBUIÇÃO = Concatenação + Cisão



Remoção em Árvore B

1. A chave se encontra em uma folha
A entrada é retirada
2. A chave se encontra em um nó não folha
A chave é substituída pela chave y imediatamente maior (y está em uma folha)

Problema: Se ficar menos de d chaves na página !

Soluções: Concatenação
Redistribuição

Remoção em Árvore B

Concatenação

P e Q são irmãs adjacentes se possuem o mesmo pai W e são apontadas por ponteiros consecutivos.

Se P e Q juntas possuírem menos de $2d$ chaves, elas podem ser concatenadas.

P e Q são agrupadas em uma página. Para isso a entrada que se encontra no pai W entre P e Q, passa também a fazer parte da nova página.

A nova página possui $2d$ entradas.

Remoção em Árvore B

Concatenação

A concatenação é propagável e pode alterar a altura da árvore.

Remoção em Árvore B

Redistribuição

P e Q são irmãs adjacentes e possuem juntas 2d ou mais chaves.

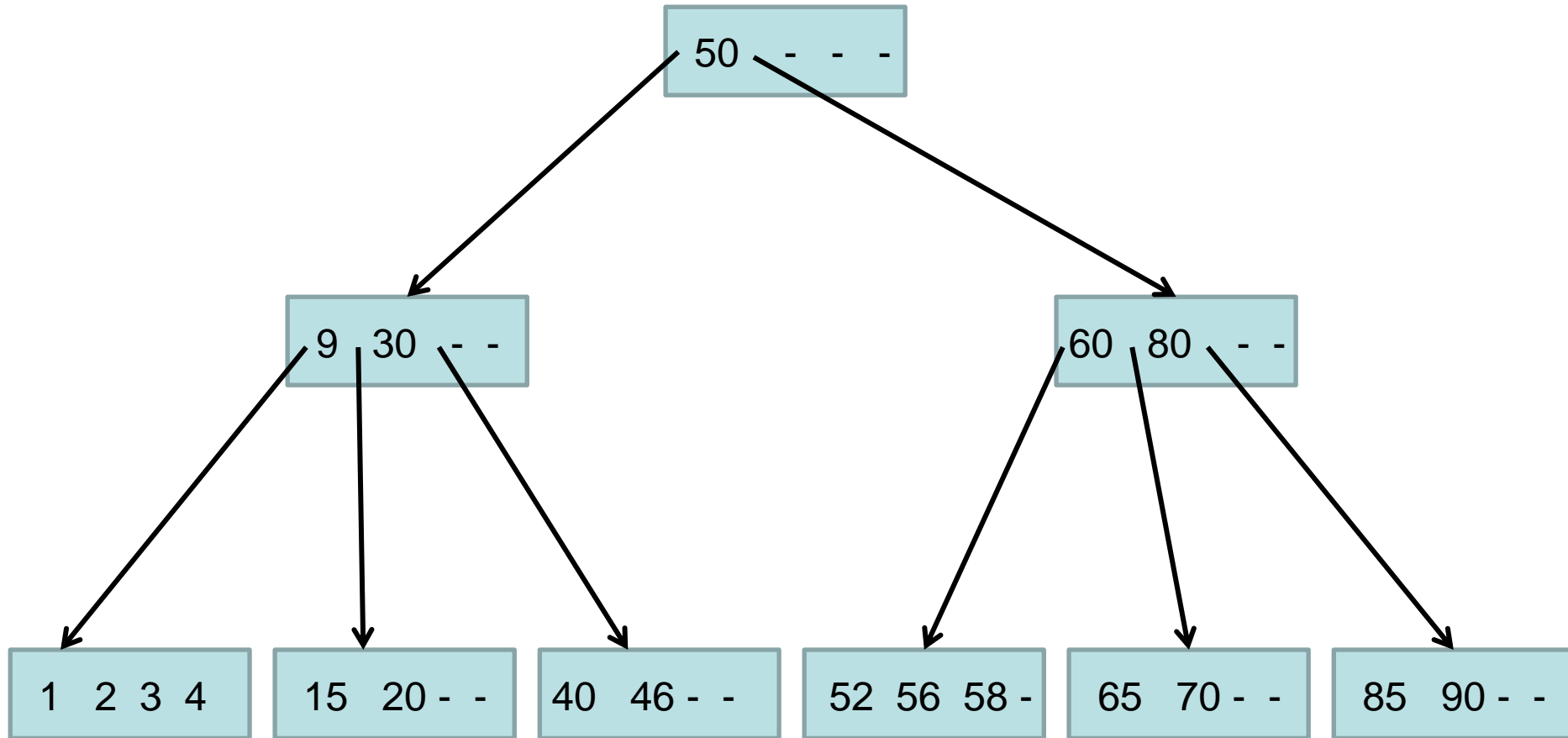
Concatenar P e Q.

Fazer a cisão de P e Q.

OBS. A redistribuição não é propagável

Exercício

2. Considerando a árvore B de ordem 2 abaixo, remover a chave 90.



Custo de Busca, Inserção e Remoção

Considere

F_{min} – mínimo número de páginas acessadas

F_{max} – máximo número de páginas acessadas

E_{min} – mínimo número de páginas escritas

E_{max} – máximo número de páginas escritas

Custo de Busca, Inserção e Remoção

BUSCA

Custo mínimo quando se encontra a chave na raiz.

$$F_{\min} = 1$$

Custo máximo quando se encontra a chave em uma folha.

$$F_{\max} = h$$

Não há escrita de páginas

$$E_{\min} = E_{\max} = 0$$

Custo de Busca, Inserção e Remoção

INSERÇÃO

Melhor caso quando nenhuma cisão é requerida.

$$F_{\min} = h; E_{\min} = 1$$

Pior caso quando todas as páginas no caminho entre a raiz e a folha passam pelo processo de cisão.

$$F_{\max} = h; E_{\max} = 2h + 1$$

Custo de Busca, Inserção e Remoção

REMOÇÃO

Melhor caso quando nenhuma concatenação ou redistribuição são requeridas e a chave se encontra em uma folha.

$F_{min} = h$; $E_{min} = 1$

Pior caso quando ocorre concatenação em todas as páginas do caminho, exceto no dois primeiros níveis. A página filho resulta em um número insuficiente de chaves sendo necessária uma redistribuição.

Custo de Busca, Inserção e Remoção

REMOÇÃO

Há um total de $h - 2$ páginas escritas devido à concatenação e ainda 3 páginas escritas como resultado da redistribuição.

$F_{\max} = 2h - 1$ (h para baixo, $h - 1$ para cima olhando a página irmã).

$E_{\max} = h + 1$

Exercícios em Árvore B

1. Fazer o algoritmo da cisão em uma árvore B de ordem d .
 2. Fazer o algoritmo da concatenação em uma árvore B de ordem d .
 3. Fazer o algoritmo da redistribuição em uma árvore B de ordem d .
-