

INF 1010

Estruturas de Dados Avançadas

Árvores B



Árvores B

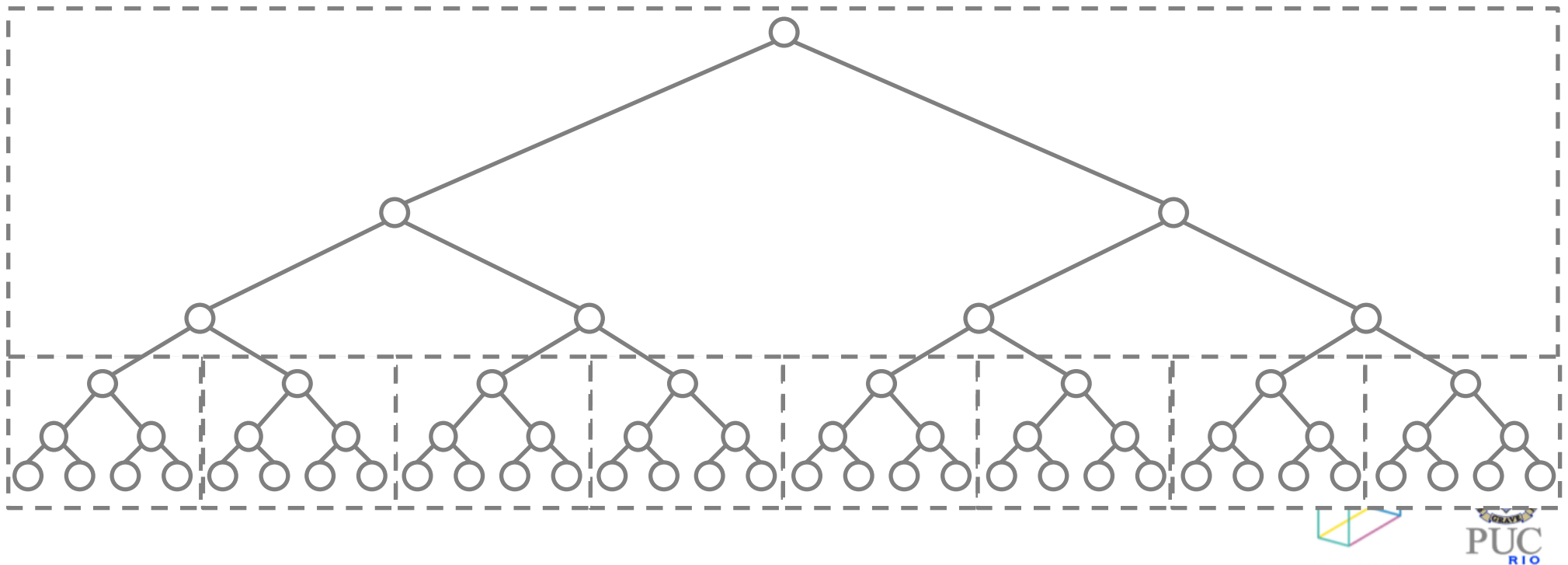
- estrutura de dados para armazenamento *externo*
- tempo de acesso à memória secundária
 - leitura em blocos
- uso frequente em sistemas de gerenciamento de bancos de dados



Árvore B - idéia

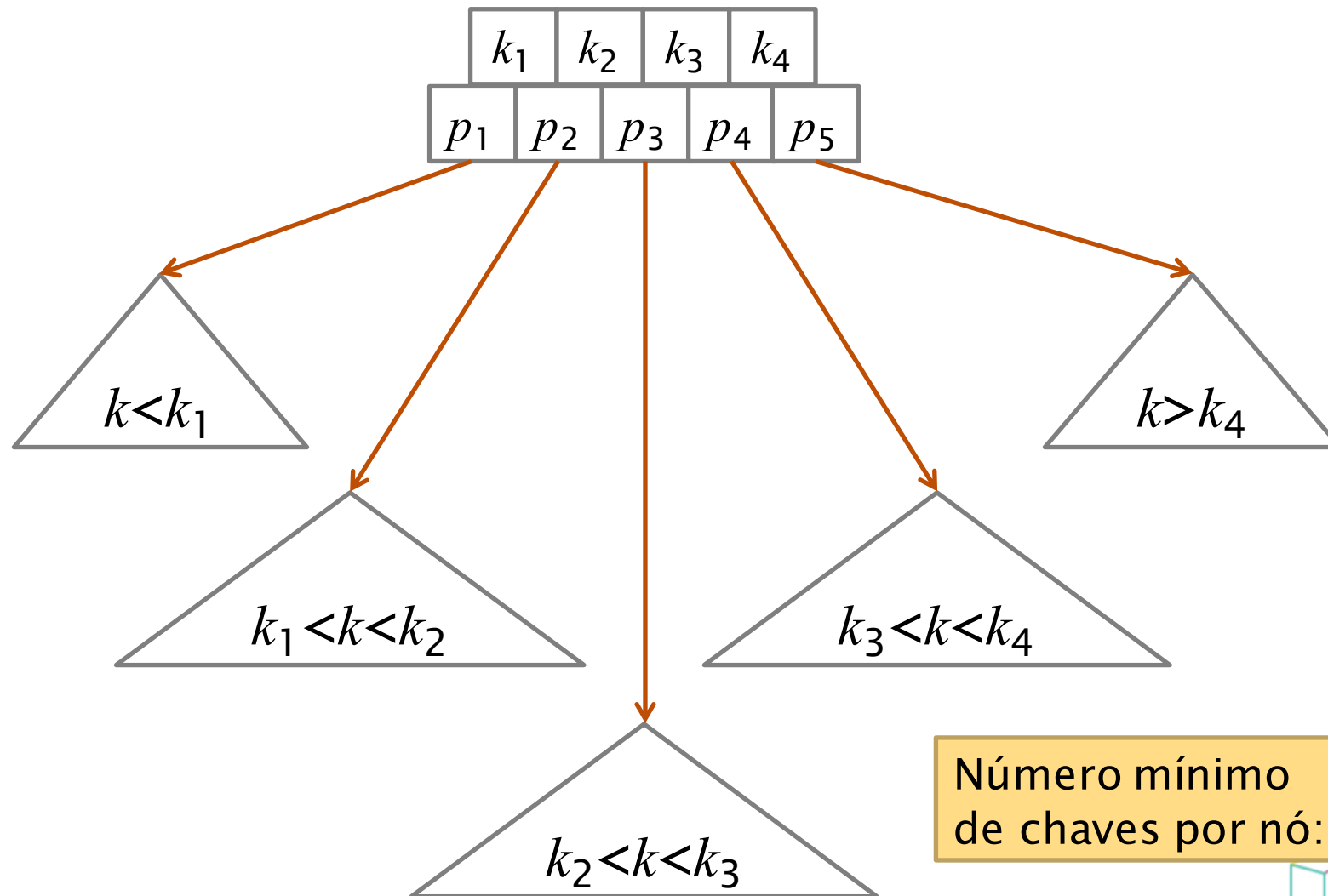
cada acesso à memória secundária traz um grupo de elementos

sub-árvores são alomeradas em **páginas**



Árvore B – idéia básica

Árvore n-ária com chaves de busca nos nós

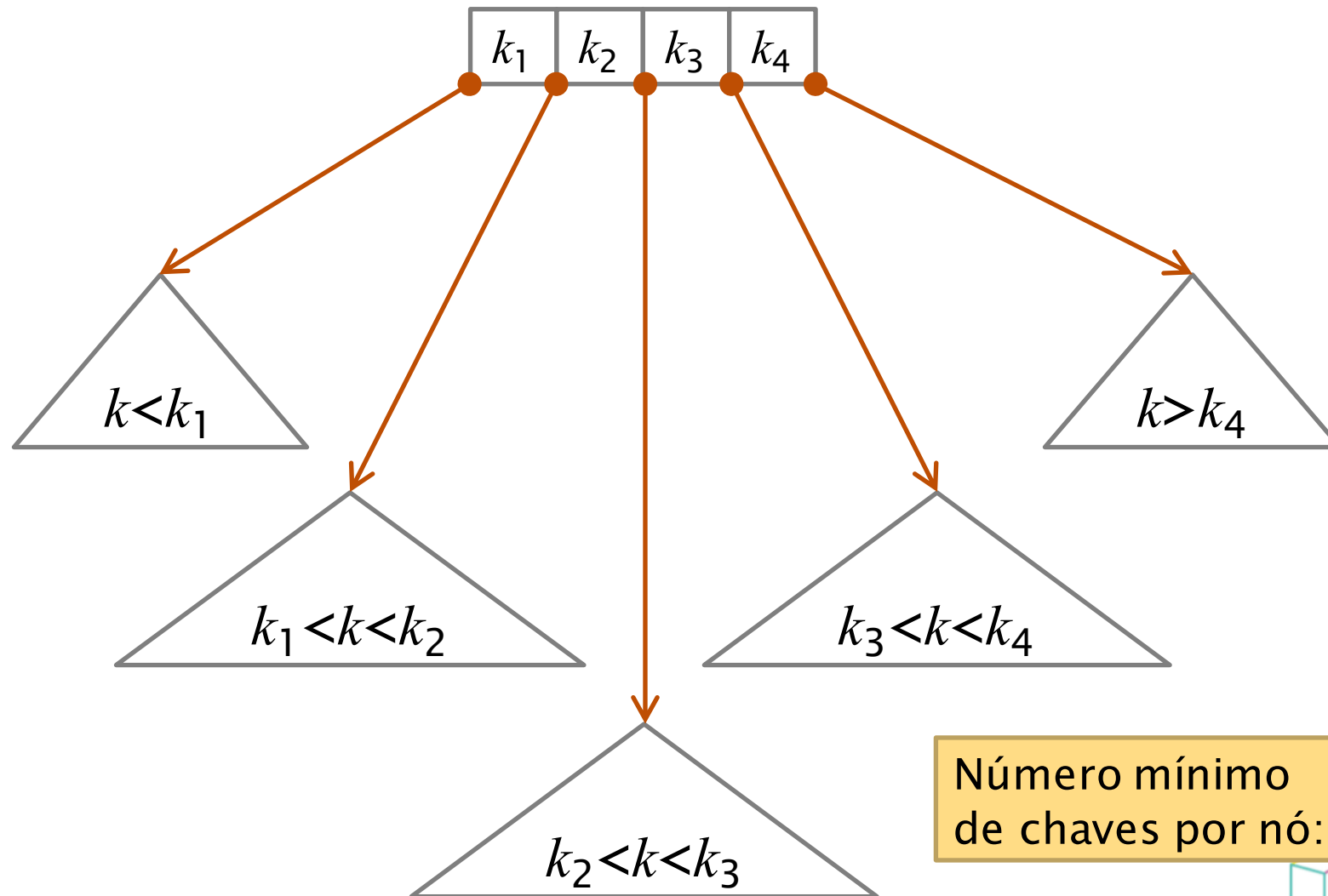


Número mínimo
de chaves por nó: $n/2$



Árvore B – idéia básica

Árvore n-ária com chaves de busca nos nós



Número mínimo
de chaves por nó: $n/2$

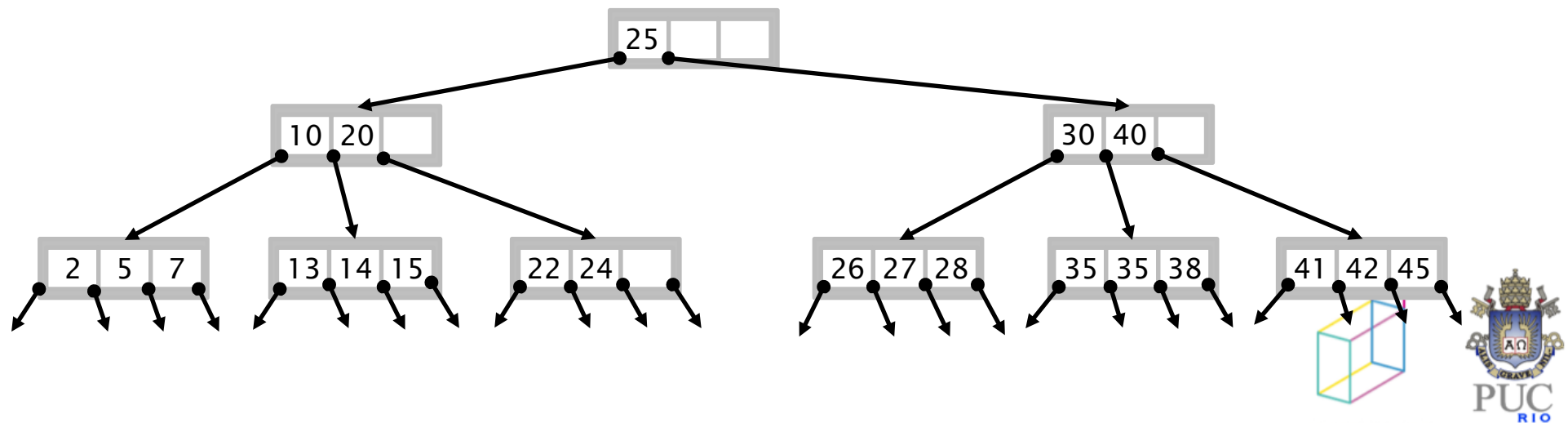


Árvore B – definição (Knuth, 1997)

árvore B de ordem m

- todo nó (página) tem no máximo m filhos

exemplo: árvore B de ordem 4 (árvore 2-3-4)

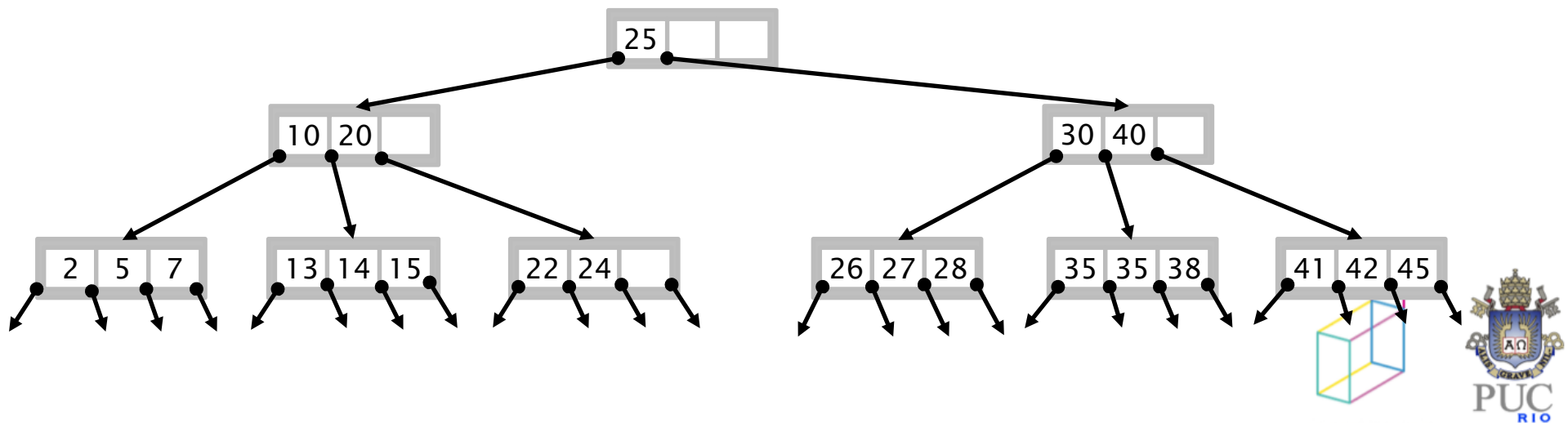


Árvore B – definição (Knuth, 1997)

árvore B de ordem m

- todo nó (página) tem no máximo m filhos
- cada nó (exceto a raiz e as folhas) possui no mínimo $\lceil m/2 \rceil$ filhos
- a raiz possui ao menos 2 filhos (a menos que seja folha)

exemplo: árvore B de ordem 4

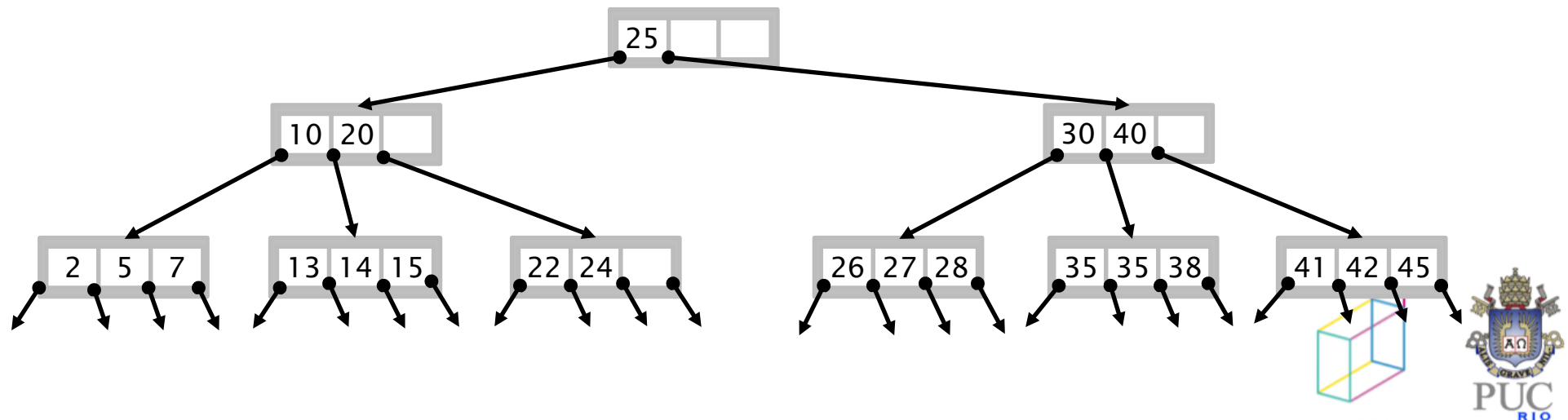


Árvore B – definição (Knuth, 1997)

árvore B de ordem m

- todo nó (página) tem no máximo m filhos
- cada nó (exceto a raiz e as folhas) possui no mínimo $\lceil m/2 \rceil$ filhos
- a raiz possui ao menos 2 filhos (a menos que seja folha)
- um nó não terminal com k filhos possui k-1 chaves
- todas as folhas aparecem no mesmo nível

exemplo: árvore B de ordem 4 (árvore 2-3-4)



Árvore B – propriedades

Número mínimo de chaves de uma Árvore B de ordem m e altura h

- A raiz tem no mínimo 1 chave e 2 filhos.
- Todos os nós internos tem no mínimo $\lceil m/2 \rceil$ filhos e $\lceil m/2 \rceil - 1$ chaves.
- Todas as folhas tem no mínimo $(\lceil m/2 \rceil - 1)$ chaves.



Árvore B – propriedades

- número mínimo de chaves de uma Árvore B de ordem m e altura h

$$\begin{aligned}k &= 1 + (\lceil m/2 \rceil - 1) * (2 + 2 * \lceil m/2 \rceil + 2 * \lceil m/2 \rceil^2 + \dots + 2 * \lceil m/2 \rceil^{(h-1)}) \\&= 1 + (\lceil m/2 \rceil - 1) * 2 * (1 + \lceil m/2 \rceil + \lceil m/2 \rceil^2 + \dots + \lceil m/2 \rceil^{(h-1)}) \\&= 1 + (\lceil m/2 \rceil - 1) * 2 * (\sum_{i=0}^{h-1} \lceil m/2 \rceil^i) \\&= 1 + (\lceil m/2 \rceil - 1) * 2 * \frac{\lceil m/2 \rceil^h - 1}{\lceil m/2 \rceil - 1} \\&= 1 + 2 * (\lceil m/2 \rceil^h - 1) \\&= 2 * (\lceil m/2 \rceil^h) - 1\end{aligned}$$



Árvore B – propriedades

- número mínimo de chaves de uma Árvore B de ordem m e altura h

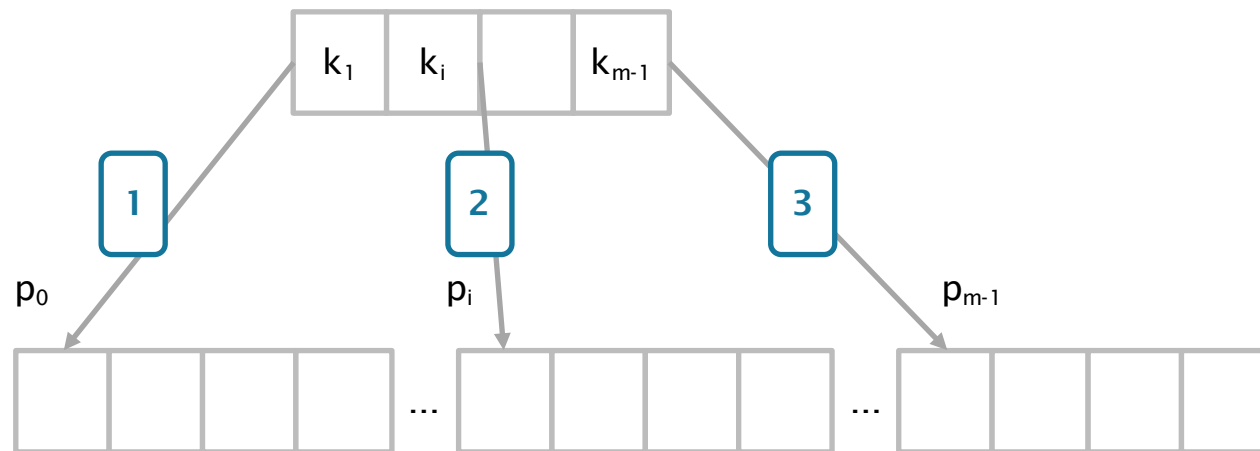
$$\begin{aligned}k &= 1 + (\lceil m/2 \rceil - 1) * (2 + 2 * \lceil m/2 \rceil + 2 * \lceil m/2 \rceil^2 + \dots + 2 * \lceil m/2 \rceil^{(h-1)}) \\&= 1 + (\lceil m/2 \rceil - 1) * 2 * (1 + \lceil m/2 \rceil + \lceil m/2 \rceil^2 + \dots + \lceil m/2 \rceil^{(h-1)}) \\&= 1 + (\lceil m/2 \rceil - 1) * 2 * (\sum_{i=0}^{h-1} \lceil m/2 \rceil^i) \\&= 1 + (\lceil m/2 \rceil - 1) * 2 * \frac{\lceil m/2 \rceil^h - 1}{\lceil m/2 \rceil - 1} \\&= 1 + 2 * (\lceil m/2 \rceil^h - 1) \\&= 2 * (\lceil m/2 \rceil^h) - 1\end{aligned}$$

$$\begin{aligned}k &\geq 2 * ((m/2)^h) - 1 \\(k+1)/2 &\geq (m/2)^h \\h &\leq \log((k+1)/2)\end{aligned}$$



Árvore B – busca

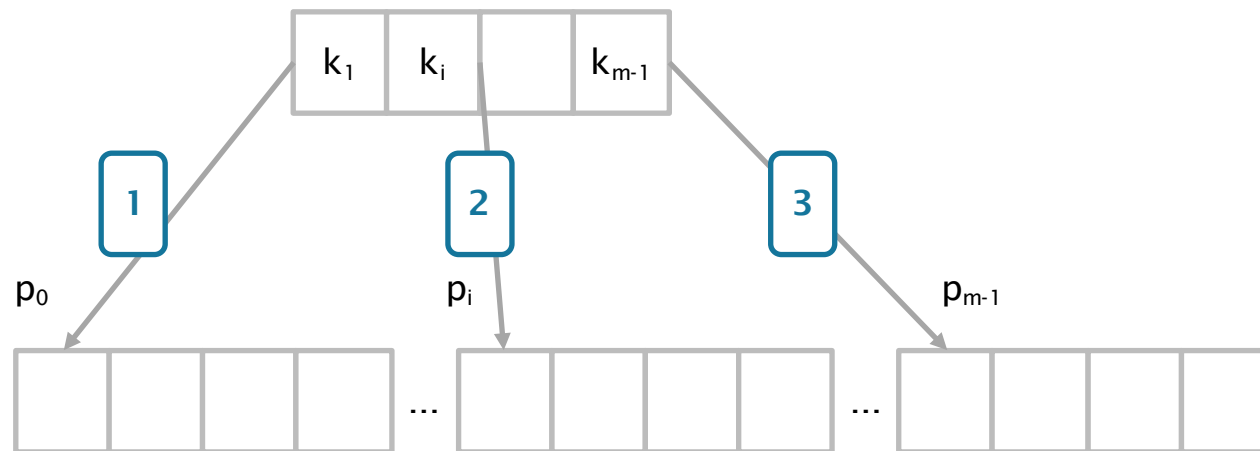
1. busca entre as chaves de uma página
2. se não for encontrada na página:
procura página para onde deve descer



Árvore B – busca

1. busca entre as chaves de uma página

$k_1 \dots k_{m-1}$ (se m for grande: busca binária)



Árvore B – busca

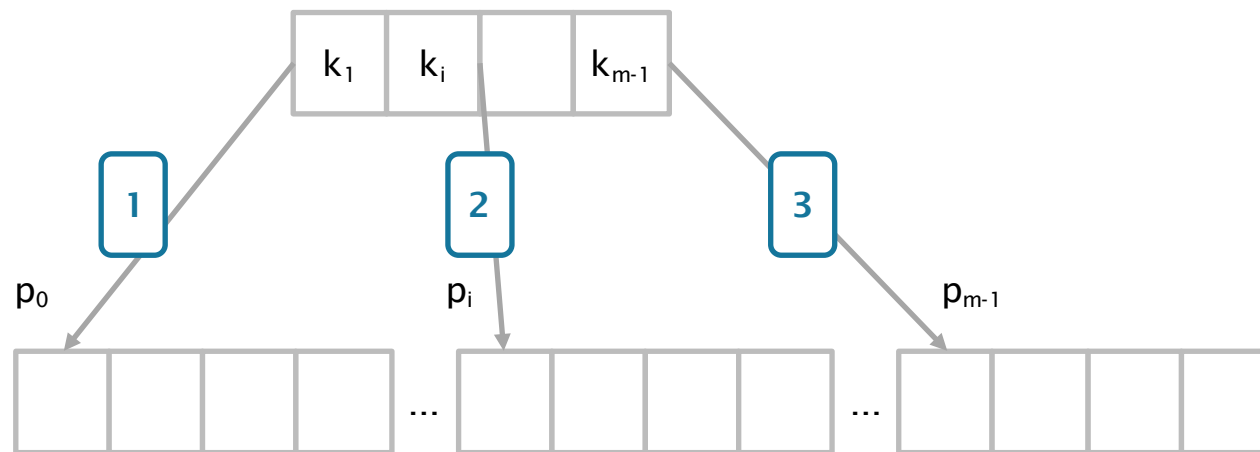
2. se não for encontrada na página:

1. $x < k_1 \rightarrow$ busca deve continuar na página p_0

2. $k_i < x < k_{i+1}$ para $1 \leq i < m-1 \rightarrow$ busca deve continuar na página p_i

3. $k_{m-1} < x \rightarrow$ busca deve continuar na página p_{m-1}

se não houver páginas abaixo da atual, a chave não existe



Árvore B de ordem m - inserção

procura chave até chegar a uma folha (se chave ã existente)

seja p_i a página onde x deverá ser inserido

se p_i tiver menos de $m-1$ elementos

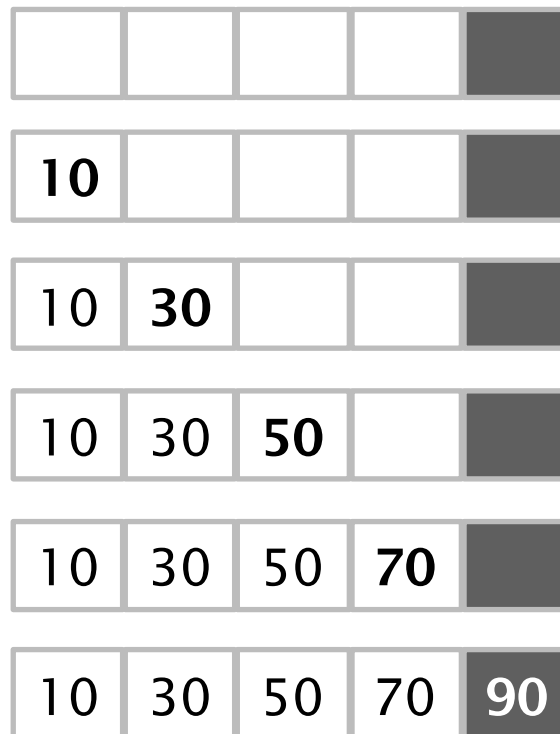
se página p_i já estiver lotada:

...



exemplo árvore ordem 5

insere 10, 30, 50, 70, 90



overflow: split



Árvore B de ordem m - inserção

seja p_i a página onde x deverá ser inserido

se p_i tiver menos de $m-1$ elementos

1. insere em p_i , na posição adequada
- se página p_i já estiver lotada...



Árvore B de ordem m - inserção

seja p_i a página onde x deverá ser inserido

...

se página p_i já estiver lotada:

1. aloca uma nova página p_k
2. distribui as m chaves da seguinte maneira:
 1. $\lceil m/2 \rceil - 1$ menores chaves em p_i
 2. $m - \lceil m/2 \rceil$ maiores chaves em p_k



Árvore B de ordem m - inserção

seja p_i a página onde x deverá ser inserido

...

se página p_i já estiver lotada:

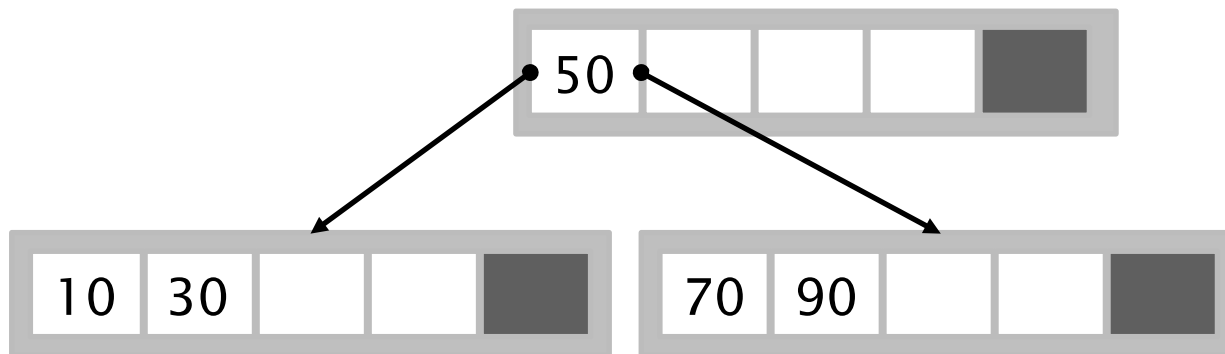
1. aloca uma nova página p_k
2. distribui as m chaves da seguinte maneira:
 1. $\lceil m/2 \rceil - 1$ menores chaves em p_i
 2. $m - \lceil m/2 \rceil$ maiores chaves em p_k
3. insere a chave mediana (em $\lceil m/2 \rceil$) na página superior
(se página p_i for raiz: cria nova raiz com a mediana)



insere 10, 30, 50, 70, 90 (ordem 5)



overflow: split

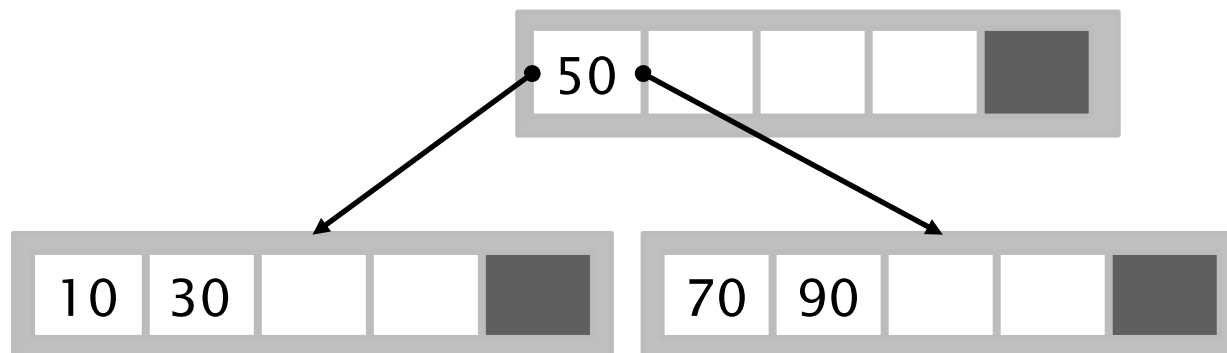


insere 10, 30, 50, 70, 90 (ordem 5)

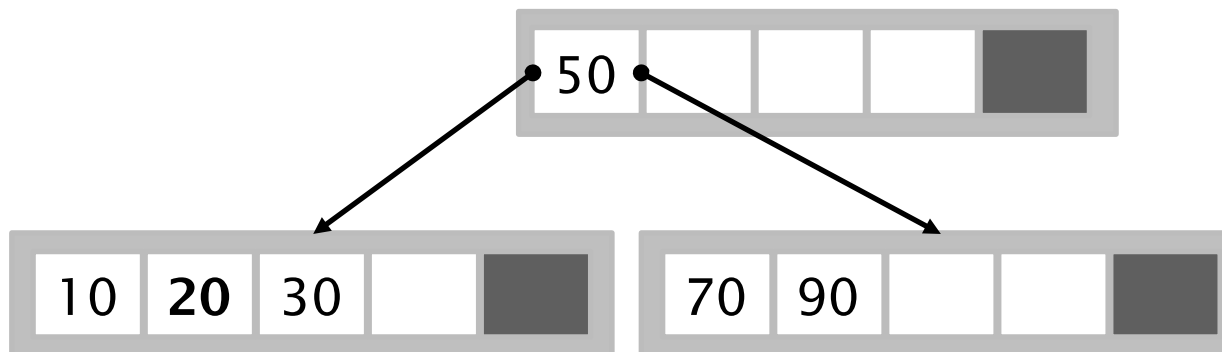
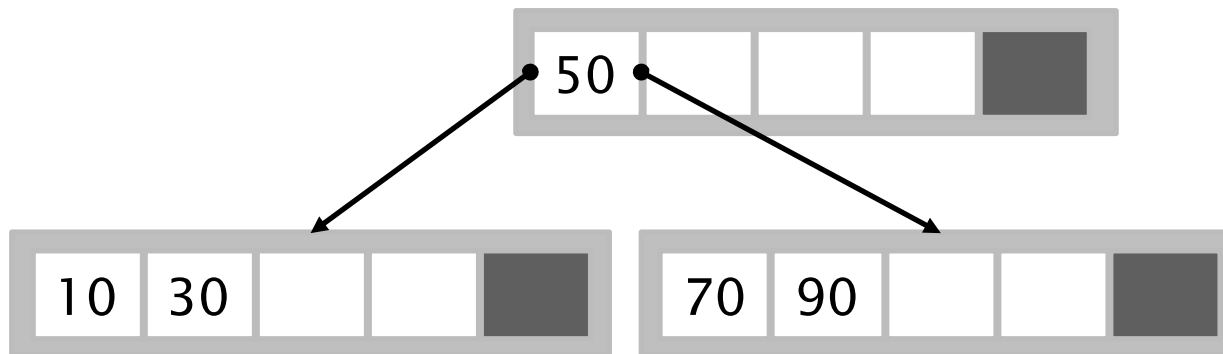


1. $\lceil m/2 \rceil - 1$ (2) menores chaves em p_i
2. $m - \lceil m/2 \rceil$ (2) maiores chaves em p_k
3. insere a chave mediana (em $\lceil m/2 \rceil$ (3)) na página superior
(se página p_i for raiz: cria nova raiz com a mediana)

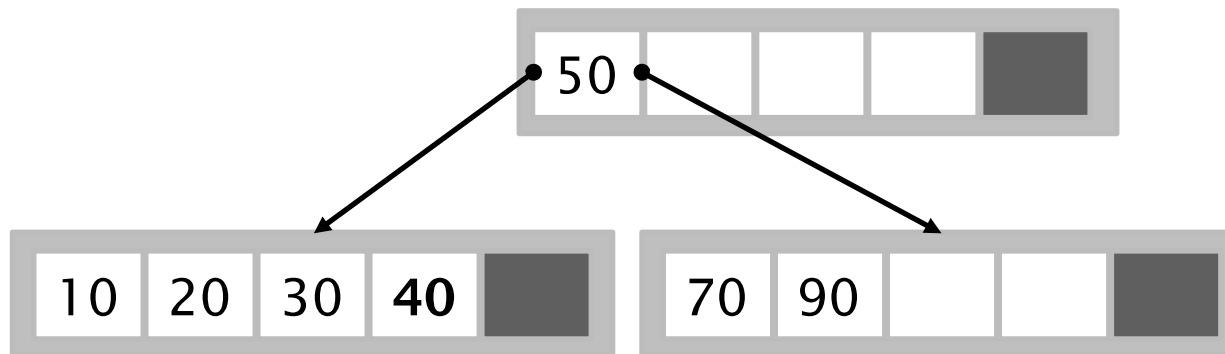
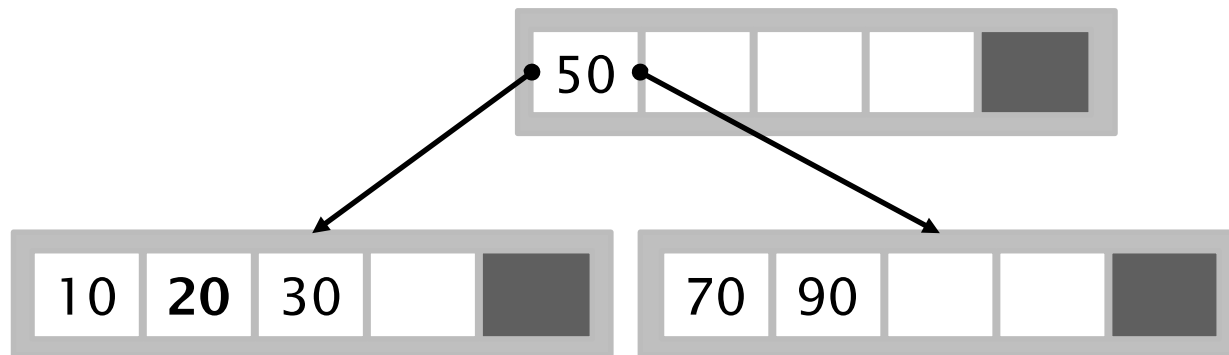
overflow: split



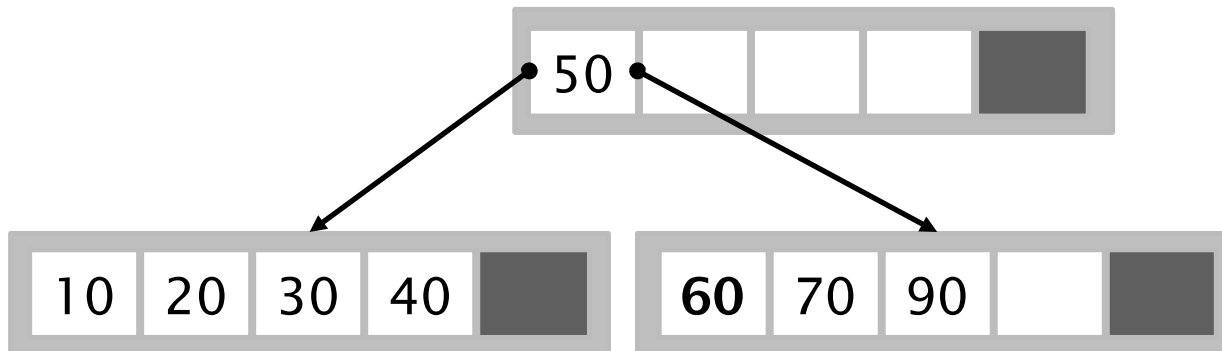
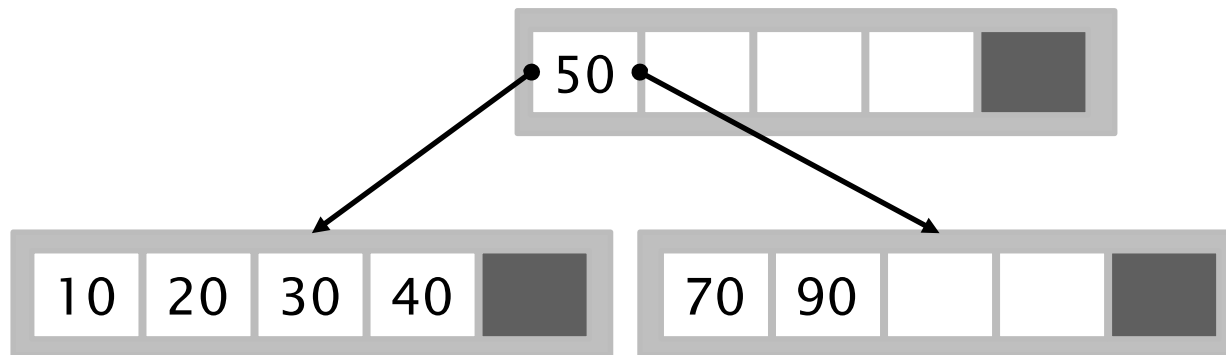
insere 20 (ordem 5)



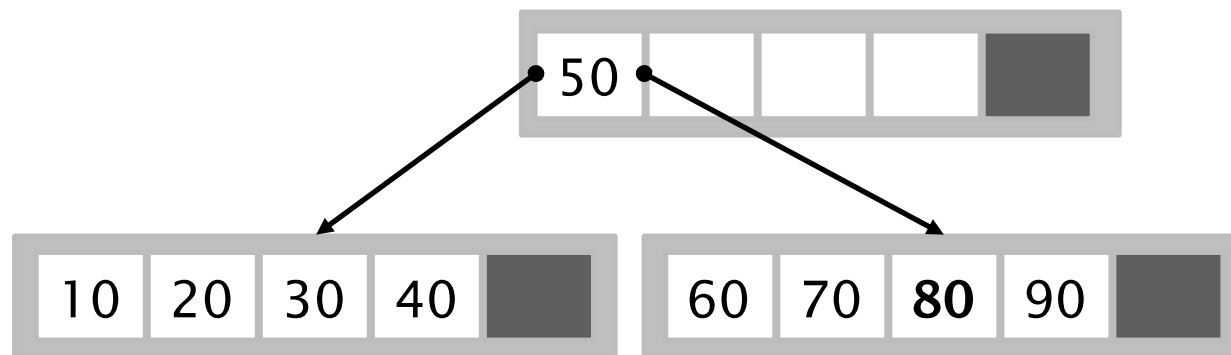
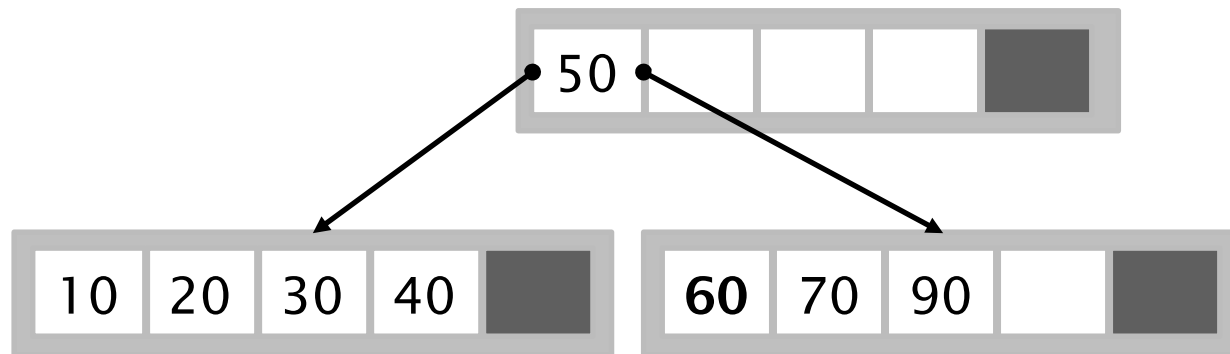
insere 40 (ordem 5)



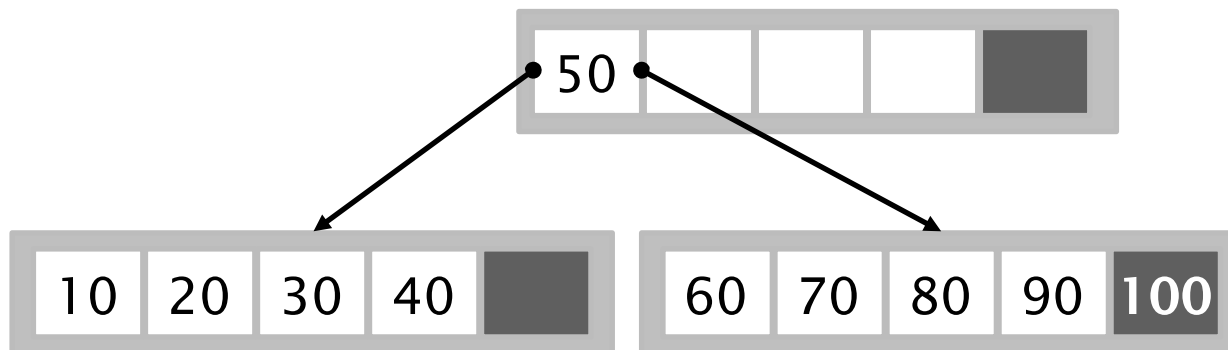
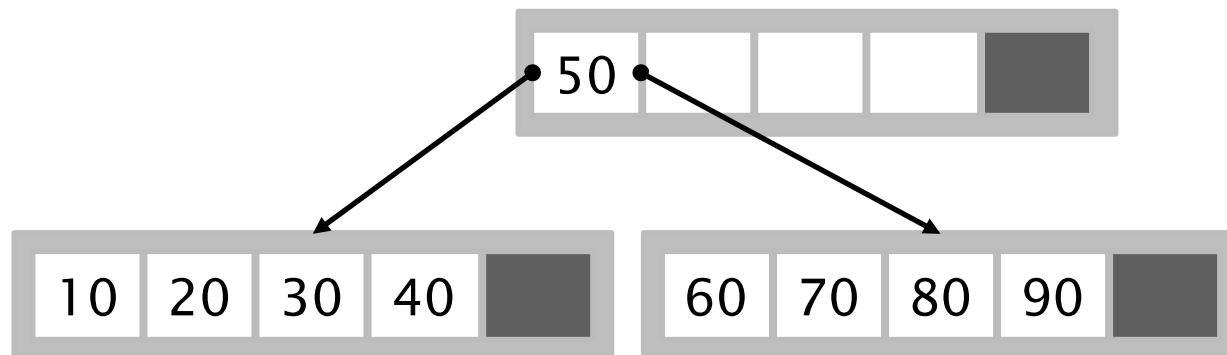
insere 60 (ordem 5)



insere 80 (ordem 5)



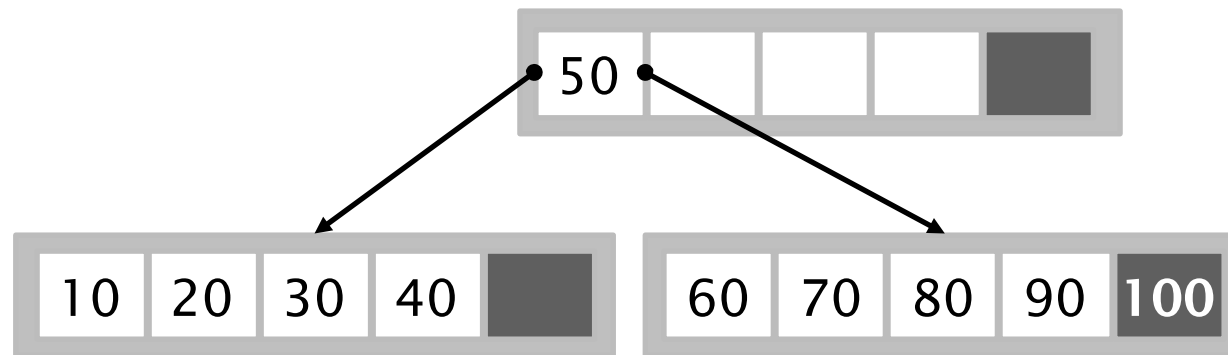
insere 100 (ordem 5)



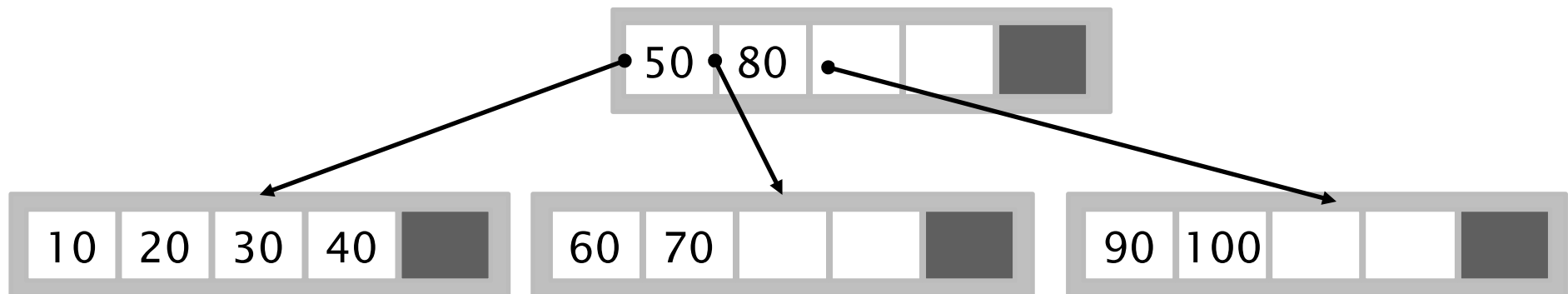
overflow: split



insere 100 (ordem 5)



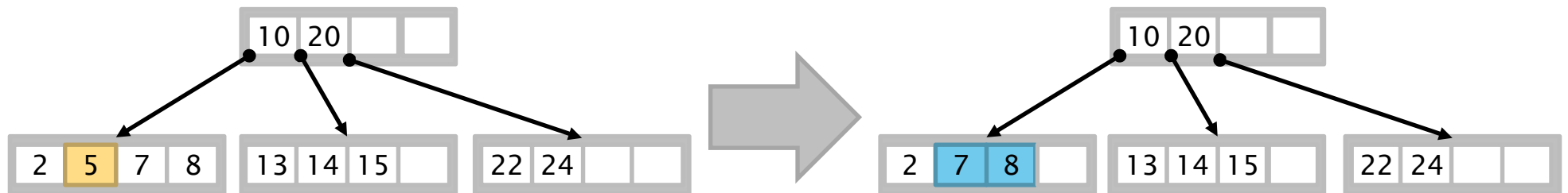
overflow: split



Árvore B de ordem m - remoção

Remoção de chave: deve ser realizada em um nó folha

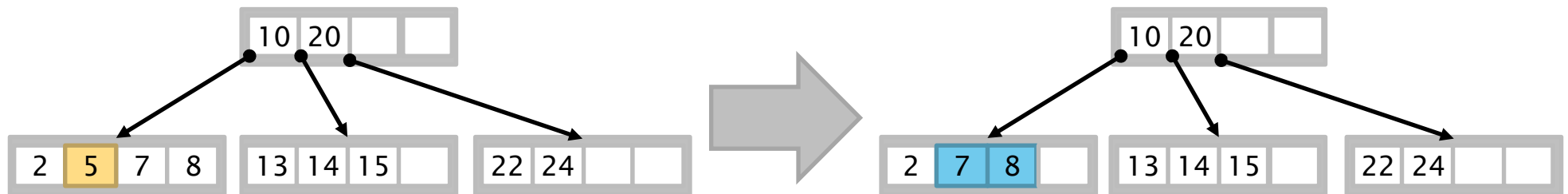
caso simples:



Árvore B de ordem m

Remoção de chave: deve ser realizada em um nó folha

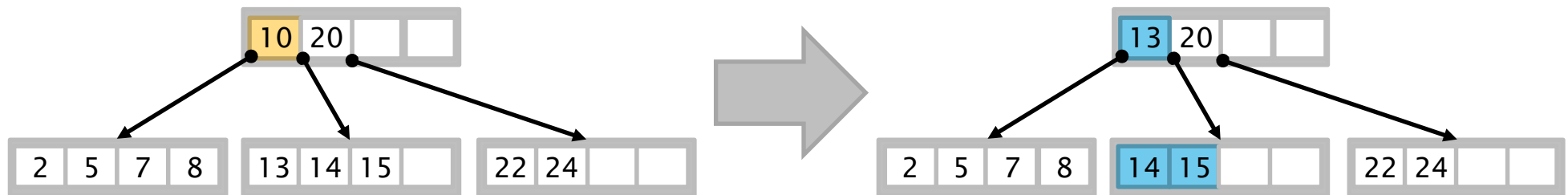
caso simples:



Árvore B de ordem m - remoção

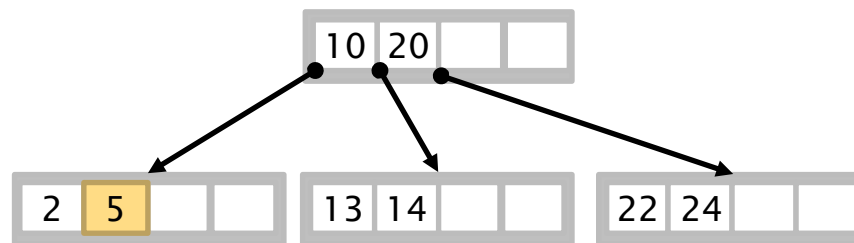
Remoção de chave: deve ser realizada em um nó folha

- Se a chave a ser removido não estiver em um nó folha, substitua-a pela maior chave da sua sub-árvore à esquerda, ou pelo menor chave da sua sub-árvore à direita.



Remoção de chave em uma folha

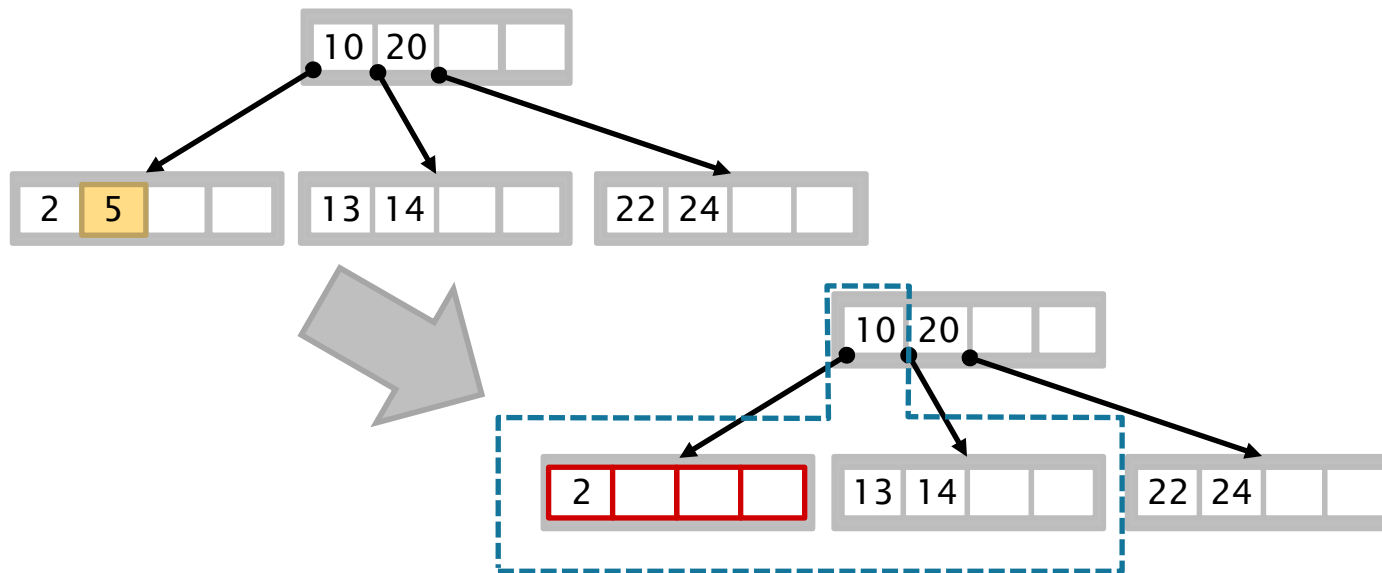
2. Quando a chave está em uma folha, deve-se removê-la e, se a folha ficar com **menos de $\lceil m/2 \rceil - 1$ chaves**, deve-se realizar uma **concatenação** ou uma **redistribuição**.



Concatenação

Concatenação:

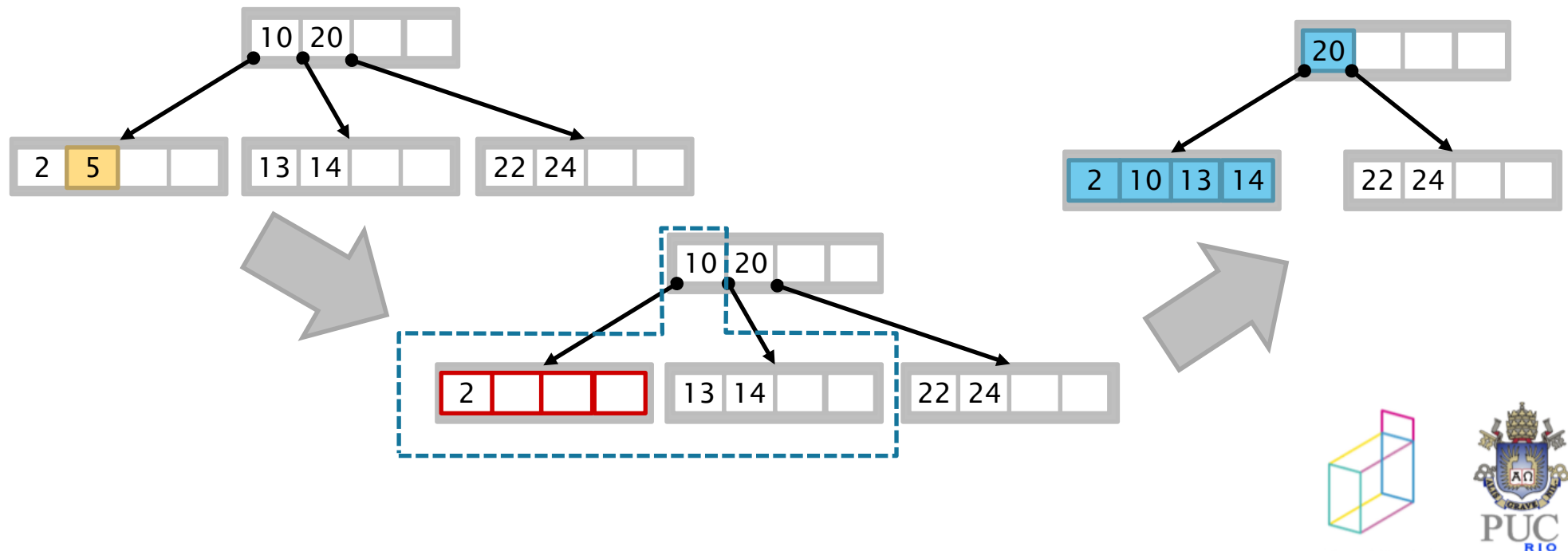
- Ocorre quando, após a remoção, a página onde a chave foi removida e uma página adjacente possuem em conjunto menos de $m-1$ chaves.



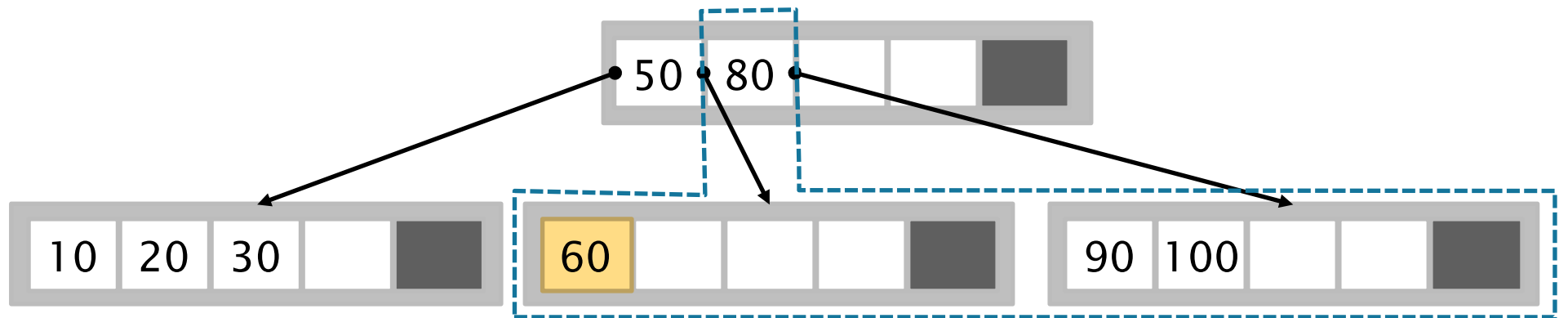
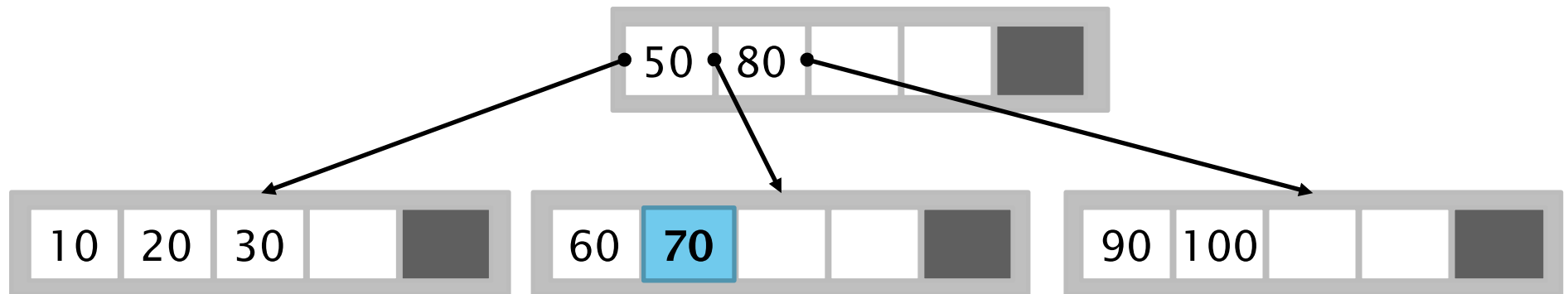
Concatenação

Concatenação:

- Ocorre quando, após a remoção, a página onde a chave foi removida e uma página adjacente possuem em conjunto com menos de $m-1$ chaves.
- Concatena-se esta página com uma adjacente. A chave do pai que estava entre elas fica na página que foi concatenada.
- Se a página resultante tiver menos do que $m/2$ chaves, repete-se o procedimento, podendo-se chegar até a raiz.



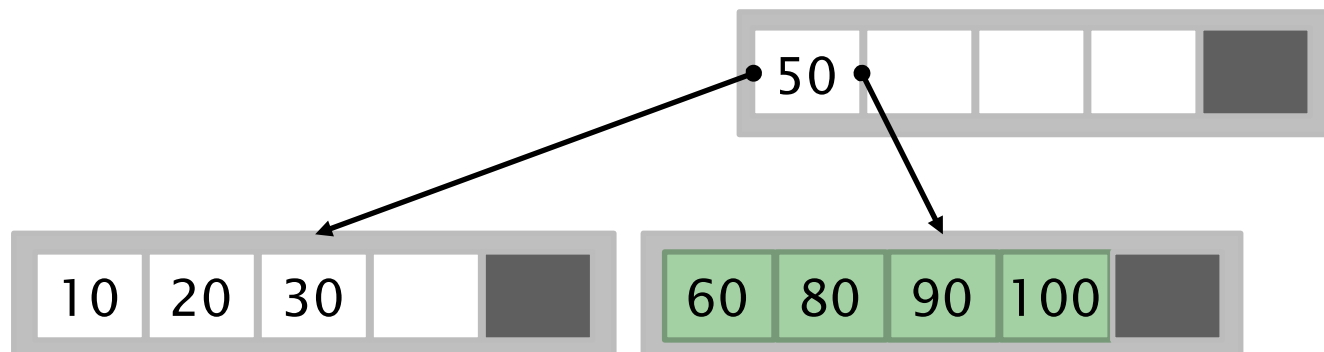
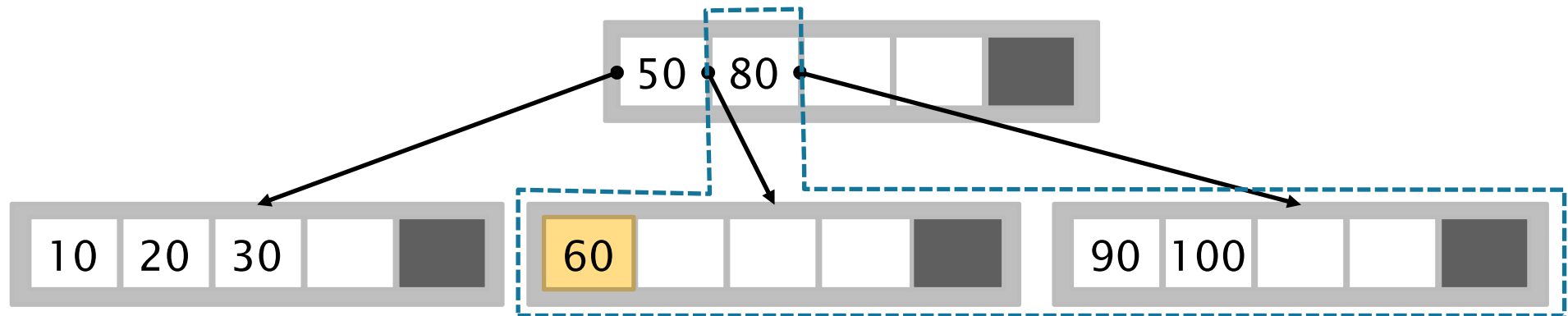
remove 70 (ordem 5)



concatena
(número de chaves < ordem)



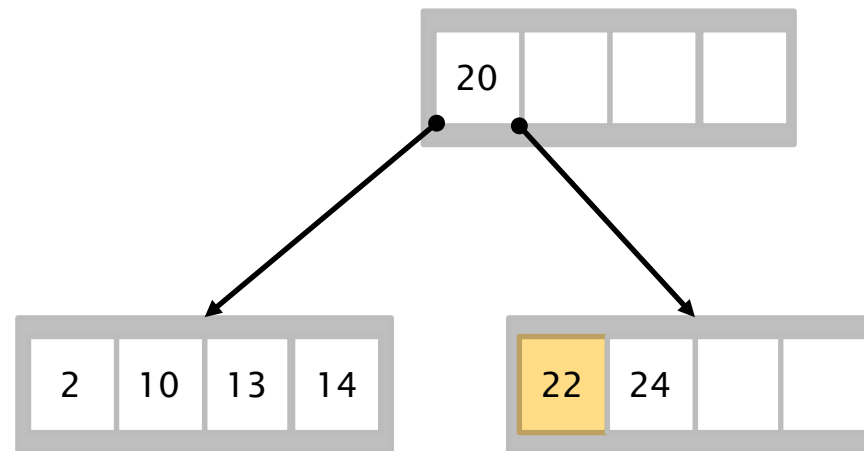
remove 70 (ordem 5)



Redistribuição

Redistribuição:

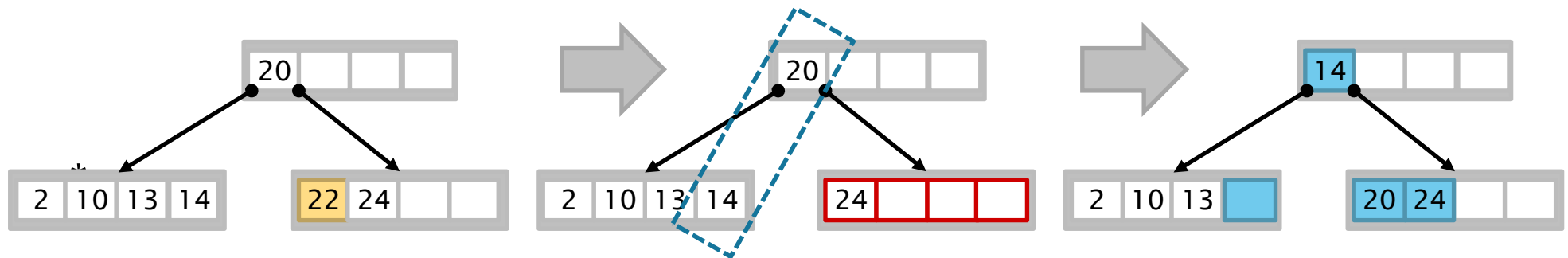
- Ocorre quando, após a remoção, a página onde a chave foi removida e uma página adjacente possuem, em conjunto, m chaves ou mais.



Redistribuição

Redistribuição:

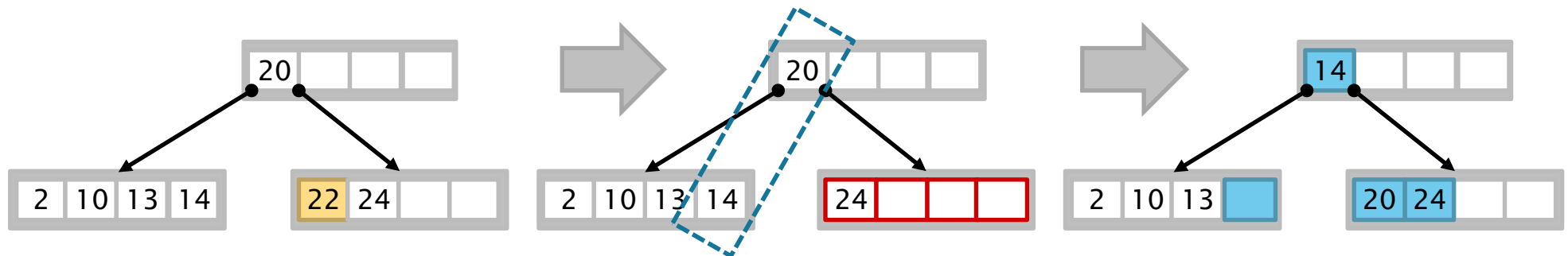
- Ocorre quando, após a remoção, a página onde a chave foi removida e uma página adjacente possuem, em conjunto, m chaves ou mais.
- Move-se a chave da página pai (“entre” as páginas adjacentes) para a página deficiente e a chave da página adjacente* para a página pai.
- Não há propagação, pois o número de chaves do pai não muda.



Redistribuição

Redistribuição:

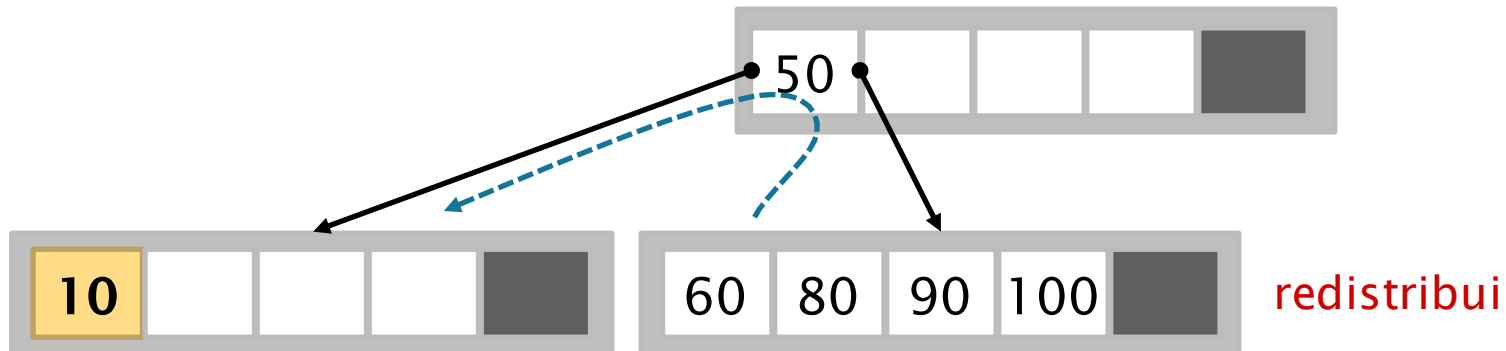
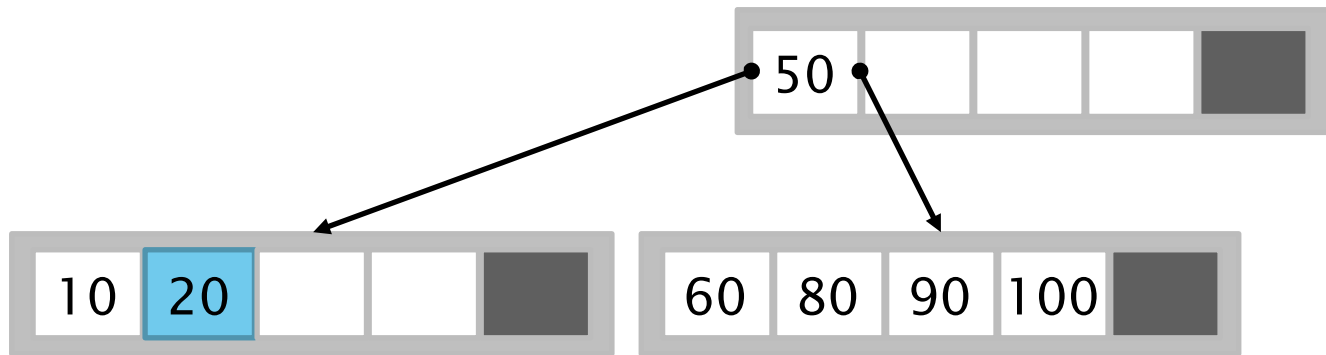
- Ocorre quando, após a remoção, a página onde a chave foi removida e uma página adjacente possuem, em conjunto, m chaves ou mais.
- Move-se a chave da página pai (“entre” as páginas adjacentes) para a página deficiente e a chave da página adjacente* para a página pai.
- Não há propagação, pois o número de chaves do pai não muda.



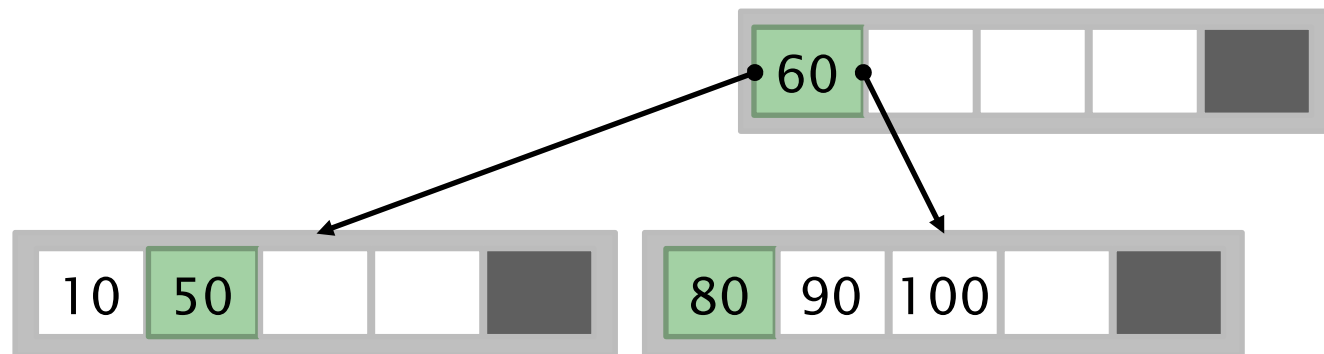
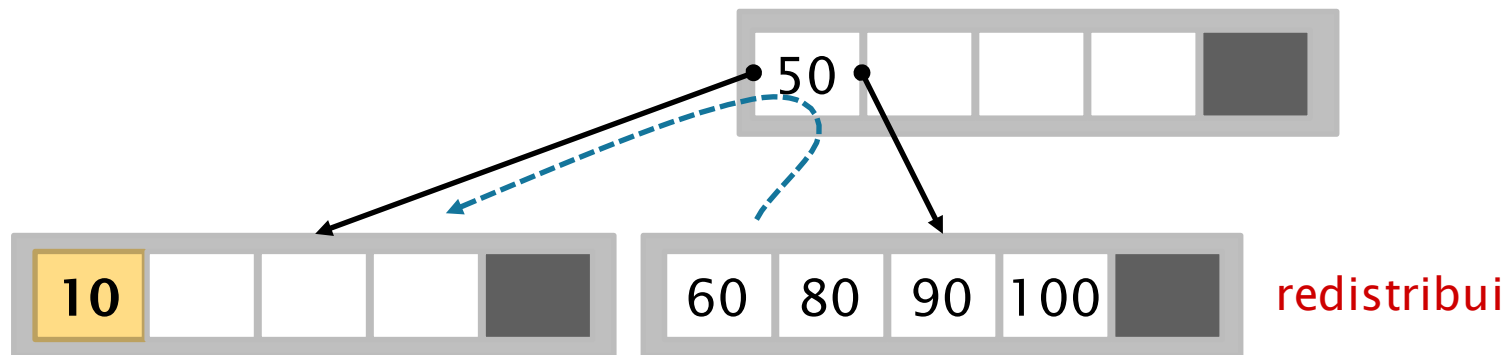
- * Se a página adjacente estiver à esquerda da página deficiente, a chave movida é a maior daquela página (*borrow from left*). Se a página adjacente estiver à direita da página deficiente, a chave movida é a menor daquela página (*borrow from right*).



remove 20 (ordem 5)



remove 20 (ordem 5)



Variantes de Árvore B

Árvore B⁺

Árvore B^{*}

...

(Outras mais recentes)

