



INF 1010

Estruturas de Dados Avançadas

Árvores B, B+ e 23

Motivação

Tornar a busca mais eficiente

considerando também os dispositivos de armazenamento de memória

Limitação da árvore binária de busca

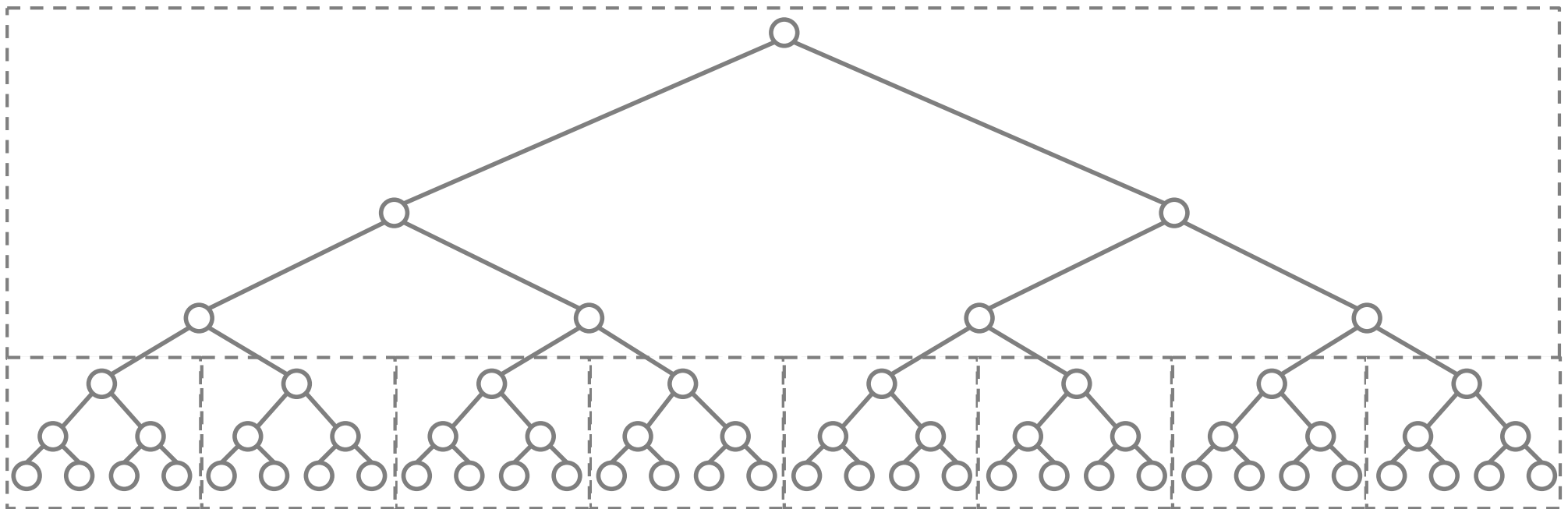
cada nó é lido individualmente

(e o acesso a memória secundária é lento)

Árvore B - definição

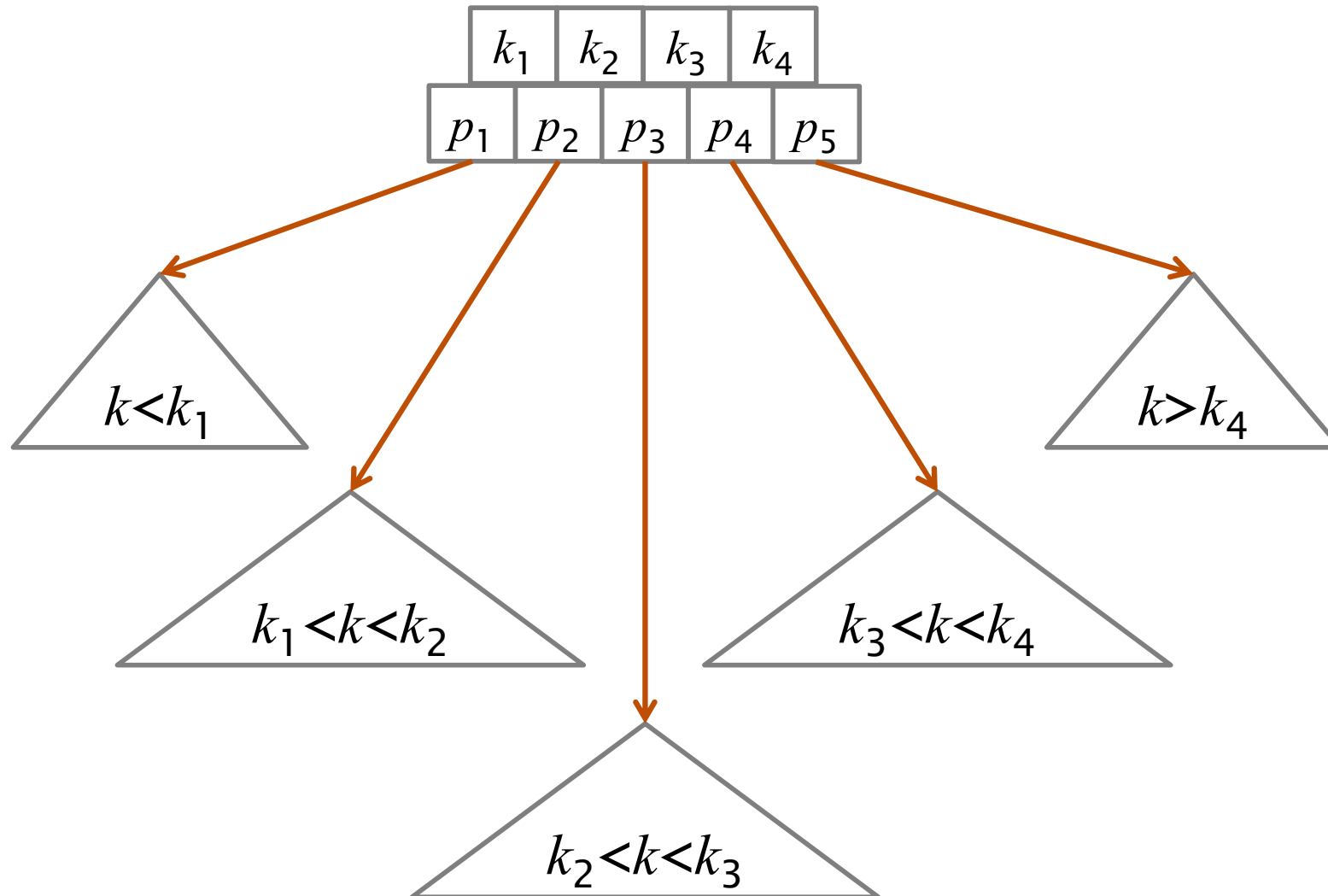
cada acesso à memória secundária traz um grupo de elementos

sub-árvores são divididas em **páginas**



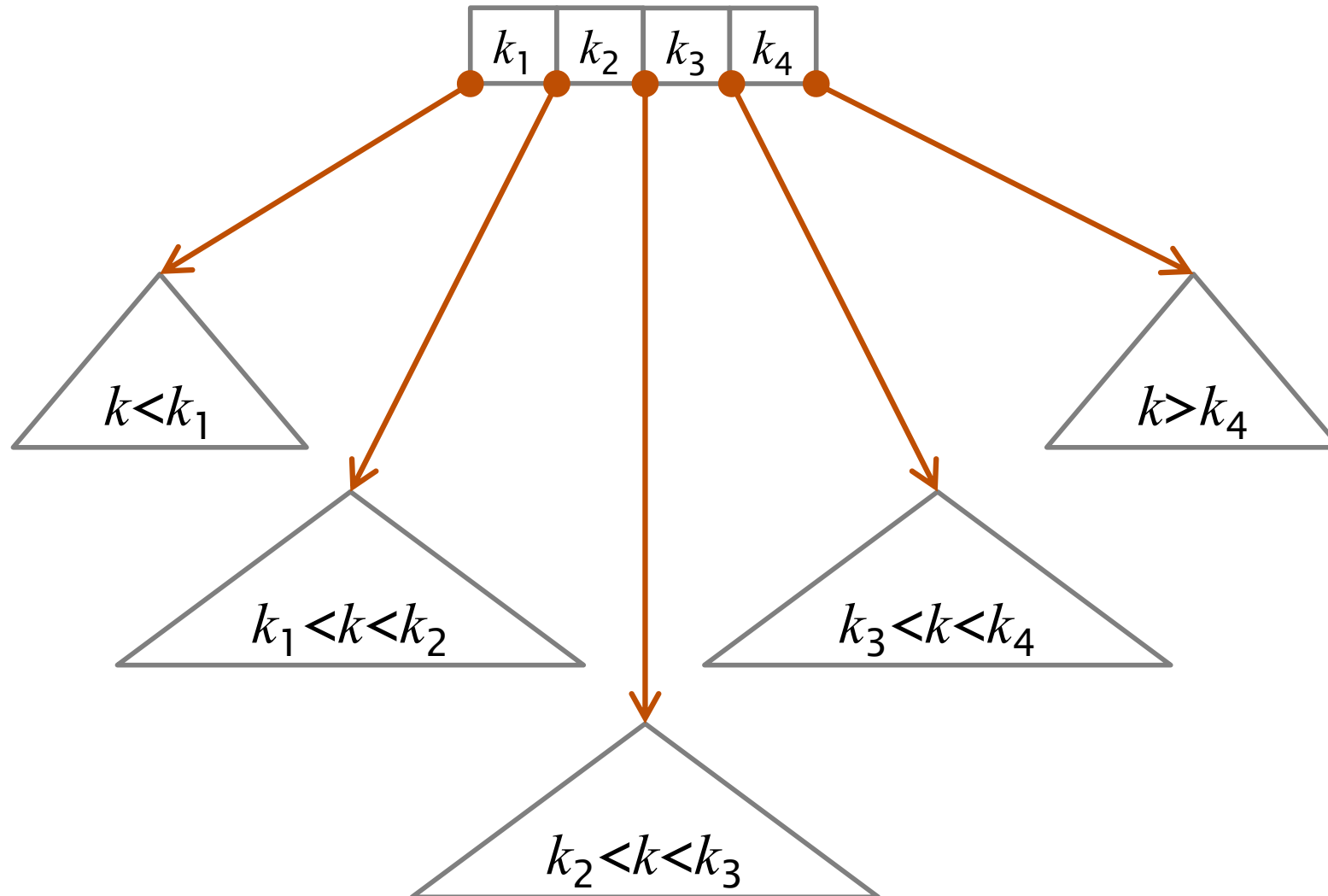
Árvore B – idéia básica

Árvore n-ária com chaves de busca nos nós



Árvore B – idéia básica

Árvore n-ária com chaves de busca nos nós

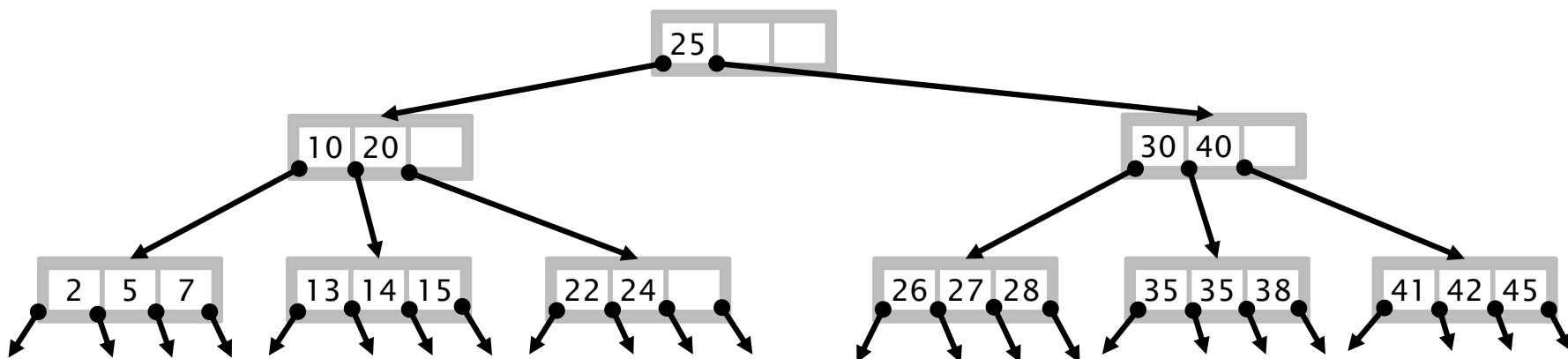


Árvore B – definição (Knuth, 1997)

árvore B de ordem m

- todo nó (página) tem no máximo m filhos
- cada nó (exceto a raiz e as folhas) possui no mínimo $m/2$ filhos
- a raiz possui ao menos 2 filhos (a menos que seja folha)
- um nó não terminal de 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 – busca

busca entre as chaves de uma página

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

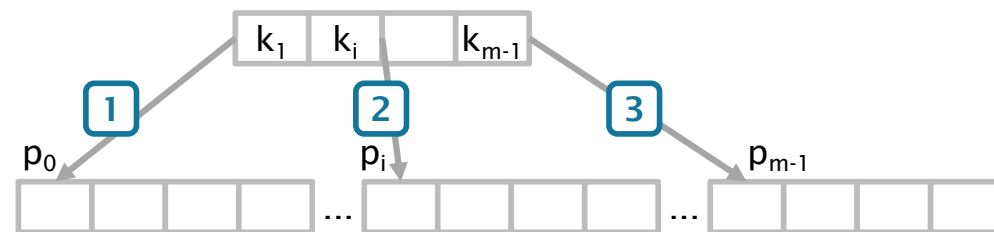
se não for encontrada na página:

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

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:

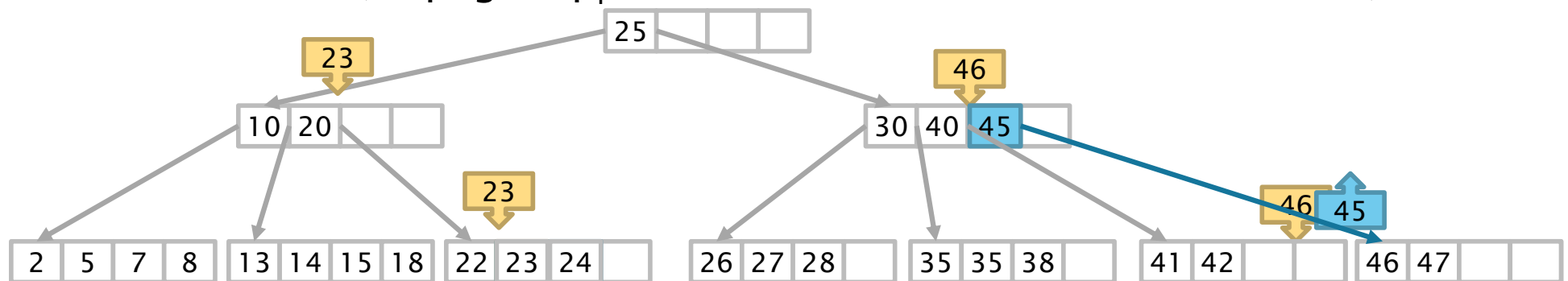
1. aloca uma nova página p_k

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

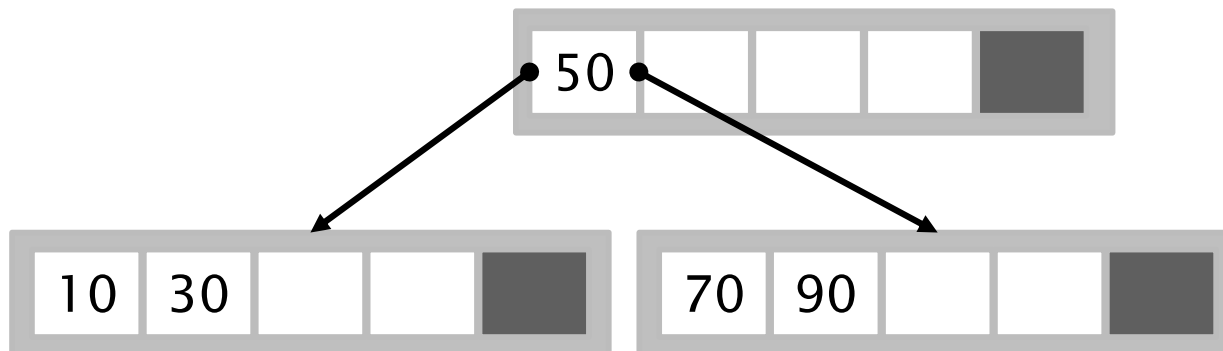
3. insere a chave mediana (em $\lceil m/2 \rceil$) na página superior
(se página p_i for raiz: cria nova raiz com a mediana)



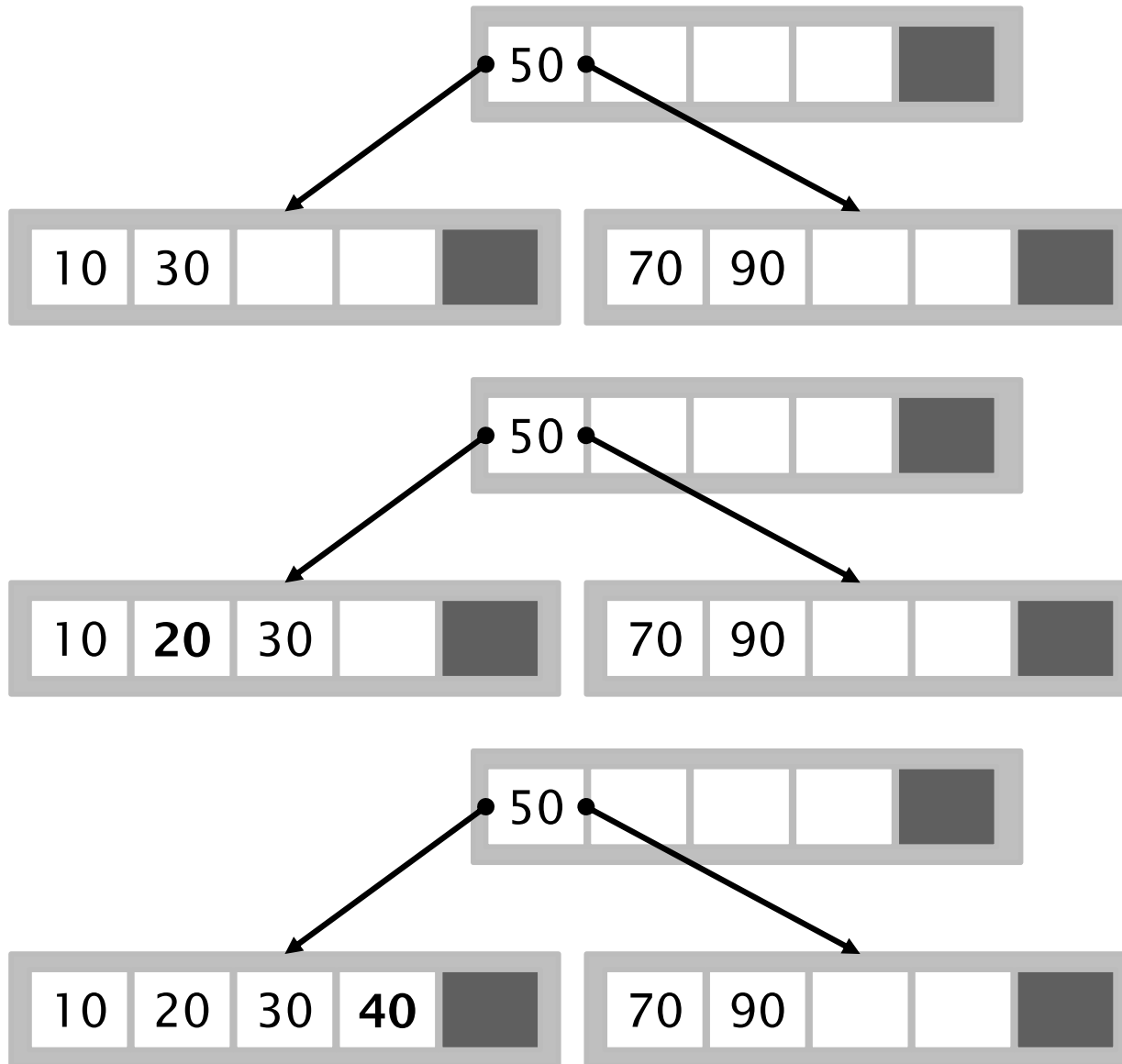
insere 10, 30, 50, 70, 90 (ordem 5)



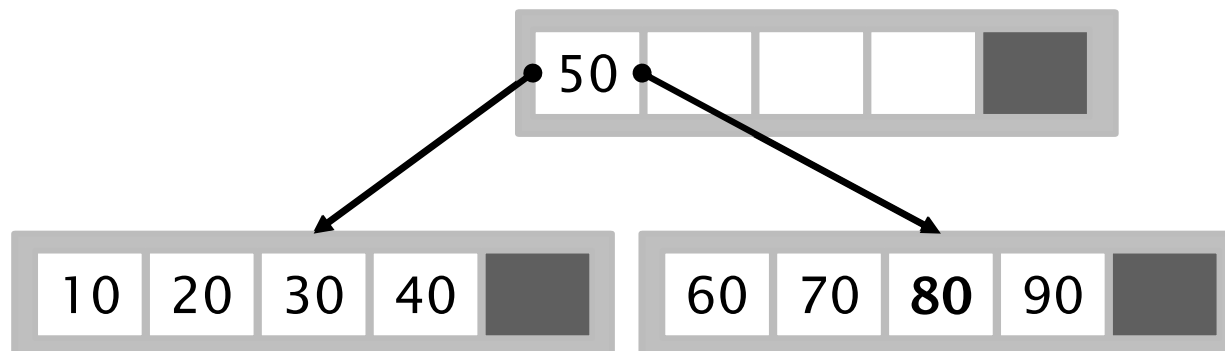
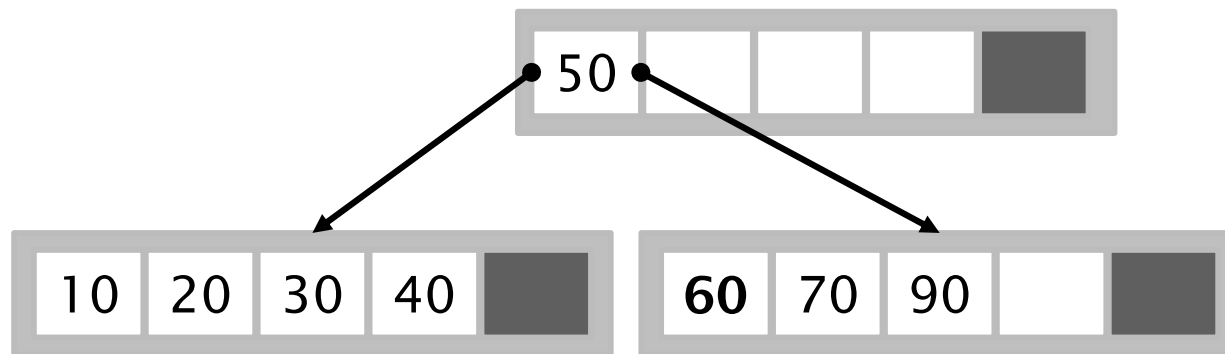
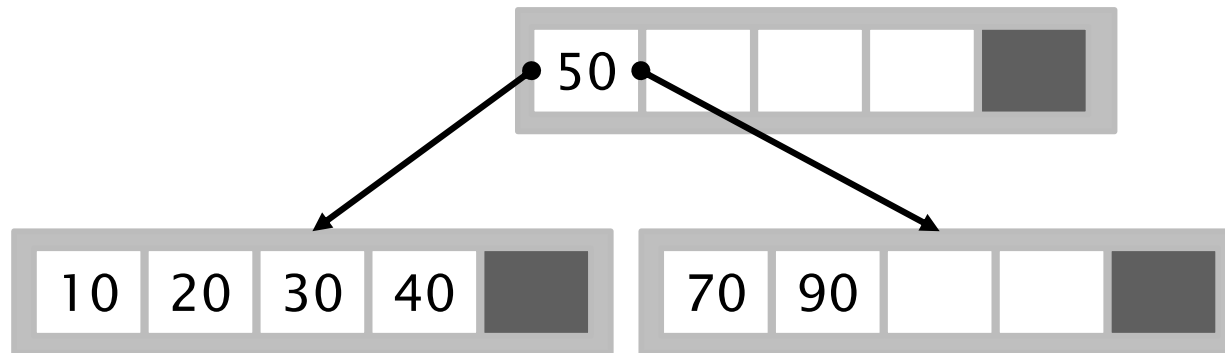
overflow: split



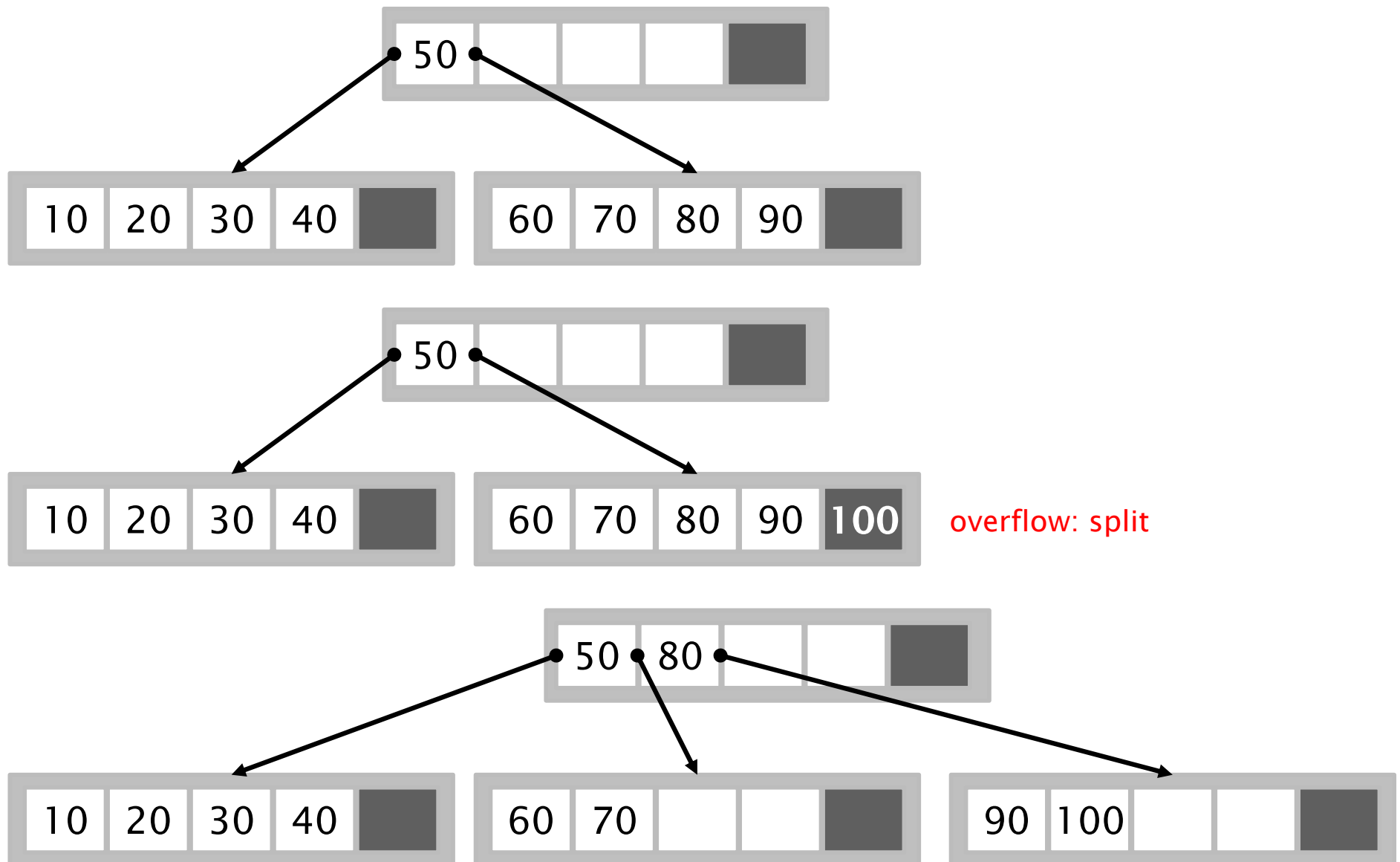
insere 20, 40 (ordem 5)



insere 60, 80 (ordem 5)



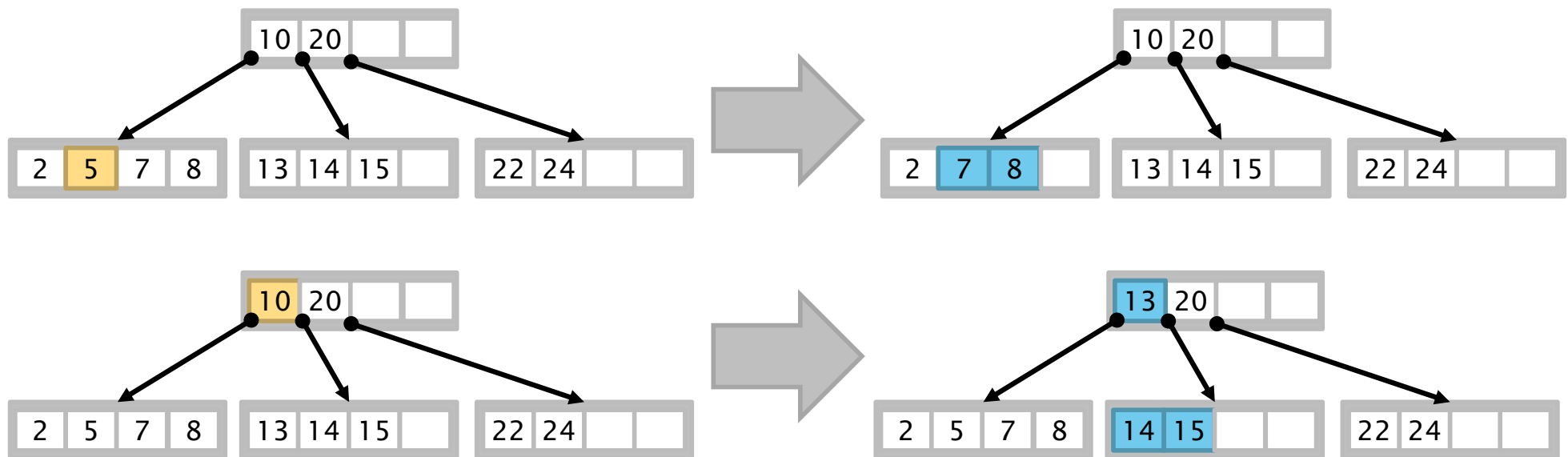
insere 100 (ordem 5)



Árvore B de ordem m - remoção

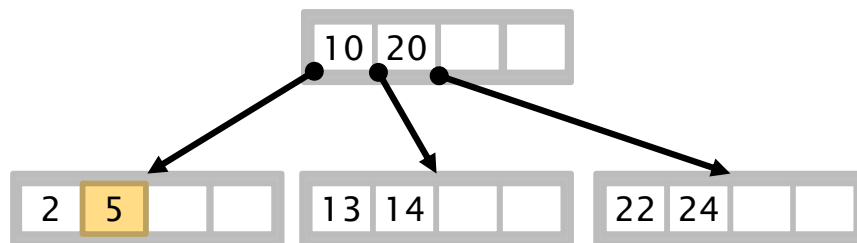
deve ser realizada em um nó folha

1. se o item a ser removido não estiver em um nó folha, substitua-o pelo maior item da sua sub-árvore à esquerda, ou pelo menor da sua sub-árvore à direita.



Remoção de chave em uma folha

- Quando a chave é uma folha, ela será removida e deverá verificar se a folha ficará com **menos de $m/2$ chaves**. Se isso acontecer, deverá ser feita uma **concatenação** ou uma **redistribuição**.

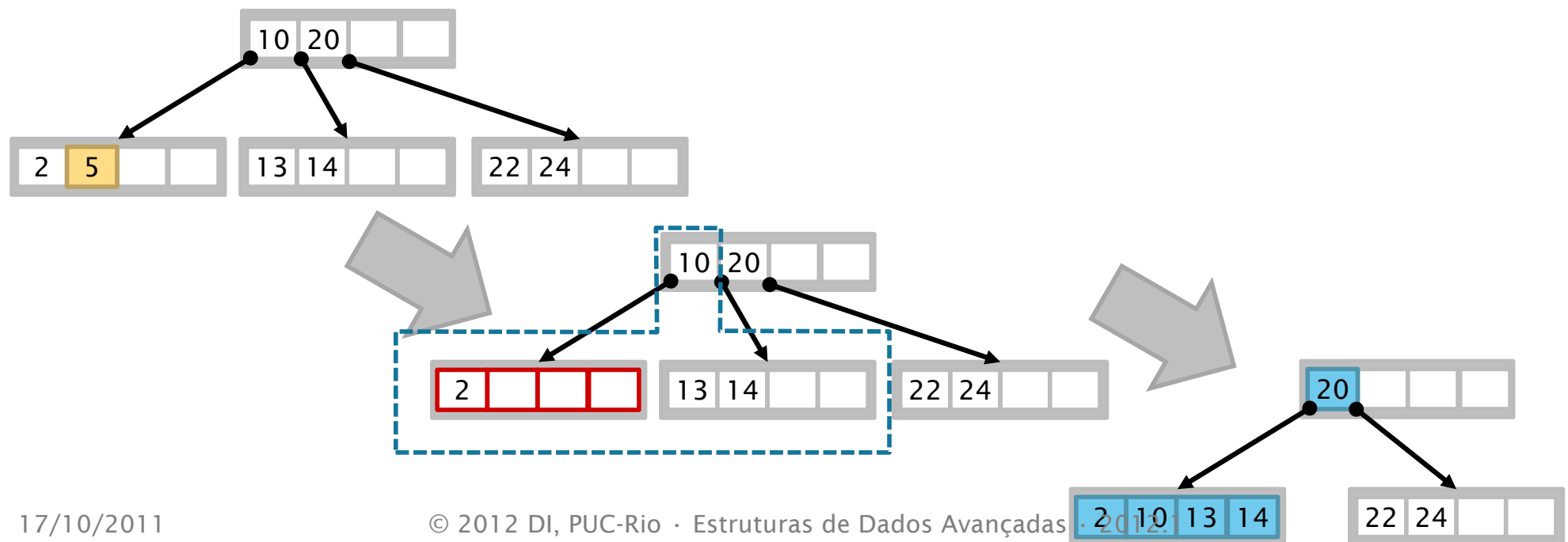


Concatenação

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

Concatene essa página com uma adjacente. A chave do pai que estava entre elas fica na página que foi concatenada.

Se esse procedimento resultar em uma página com menos de $m/2$ chaves, faça novamente o mesmo procedimento, podendo chegar até a raiz.

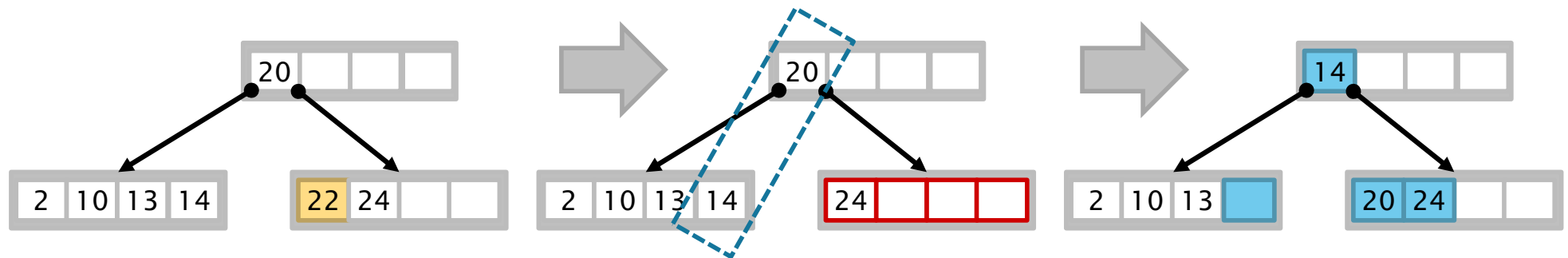


Redistribuição

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

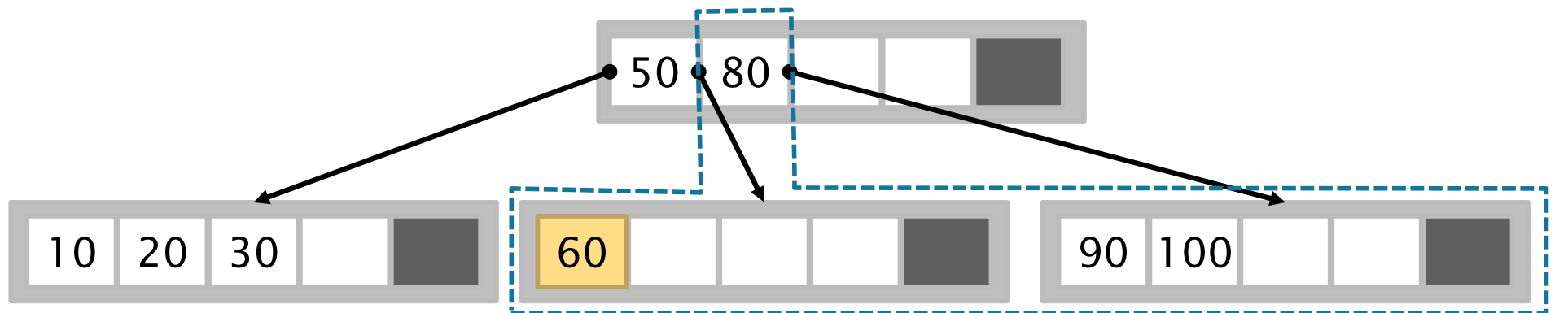
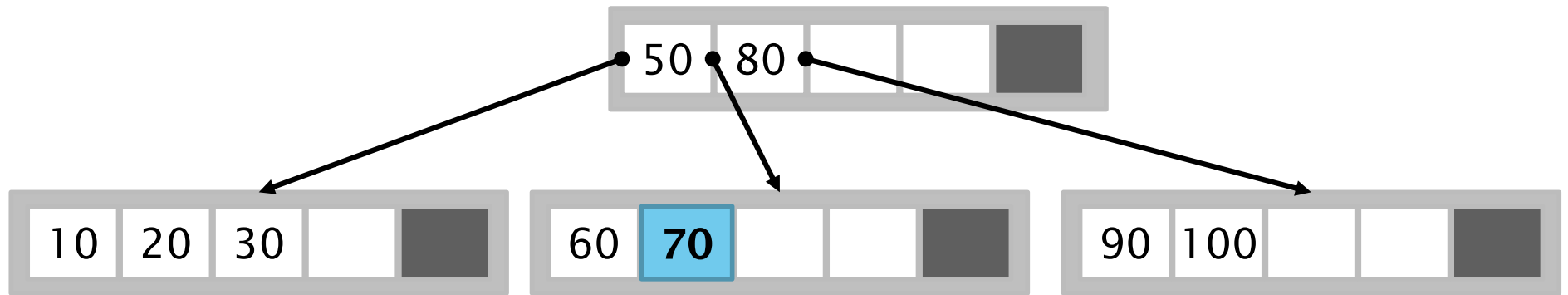
Mova a chave da página pai (“entre” as páginas adjacentes) para a página deficiente, e a chave da página adjacente* para a página pai.

Não é propagável, pois o número de chaves do pai não muda.

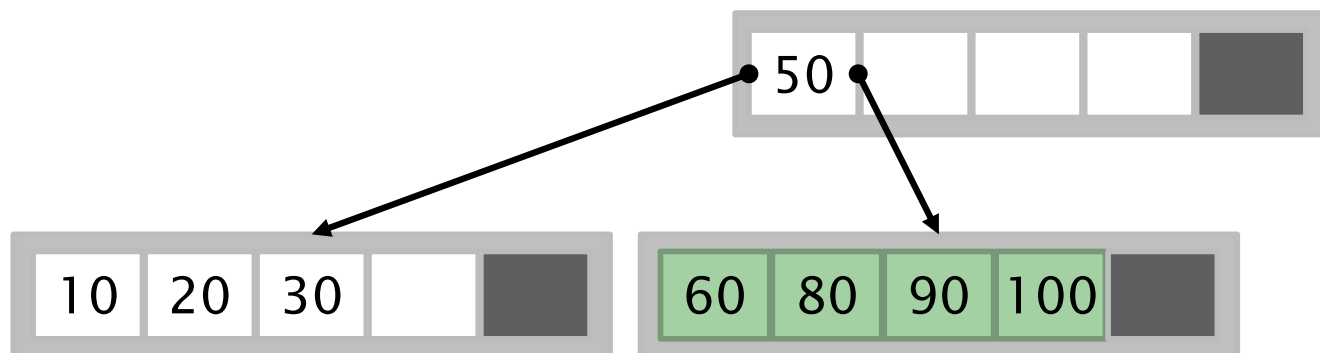


- * Se a página adjacente estiver à esquerda da página deficiente, a chave movida é a maior daquela página (*borrow from left*).
- Se a página adjacente estiver à direita da página deficiente, a chave movida é a menor daquela página (*borrow from right*).

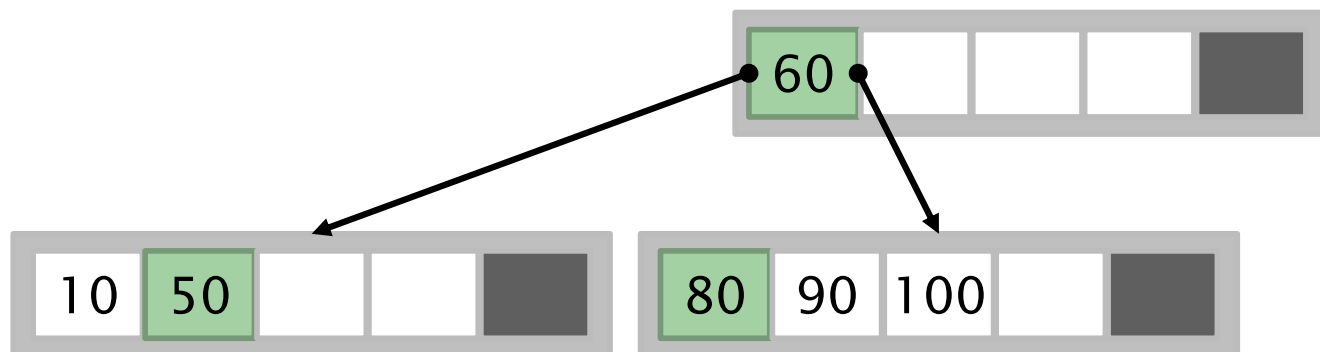
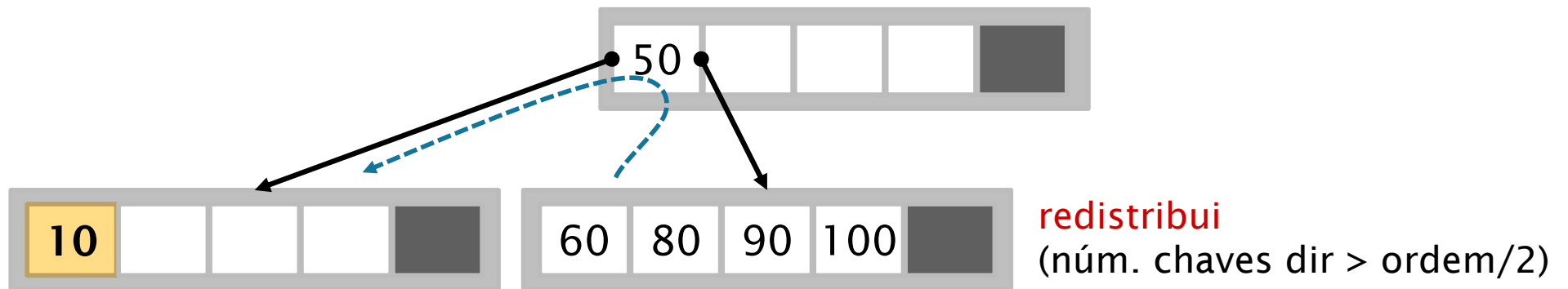
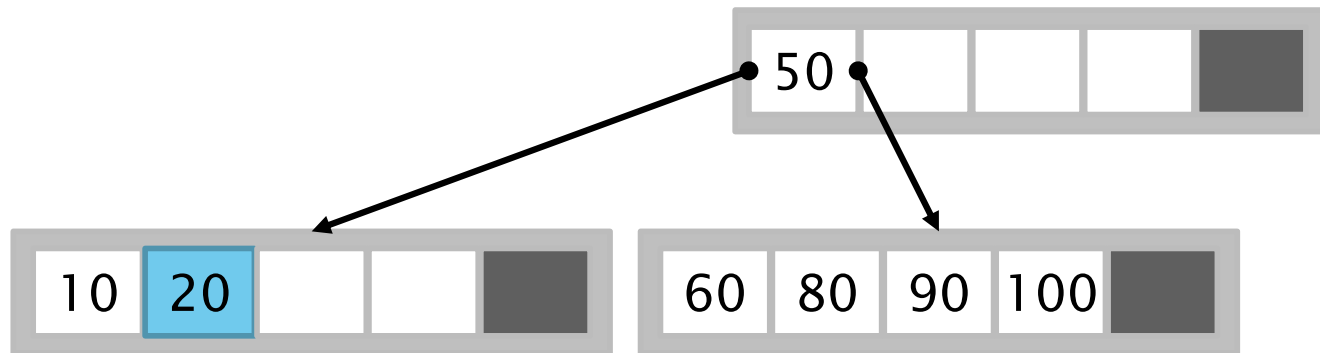
remove 70 (ordem 5)



concatena
(núm. chaves < ordem)



remove 20 (ordem 5)



árvores B+

Árvores B+ Definição

número n de filhos de um nó de ordem m

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

nós internos

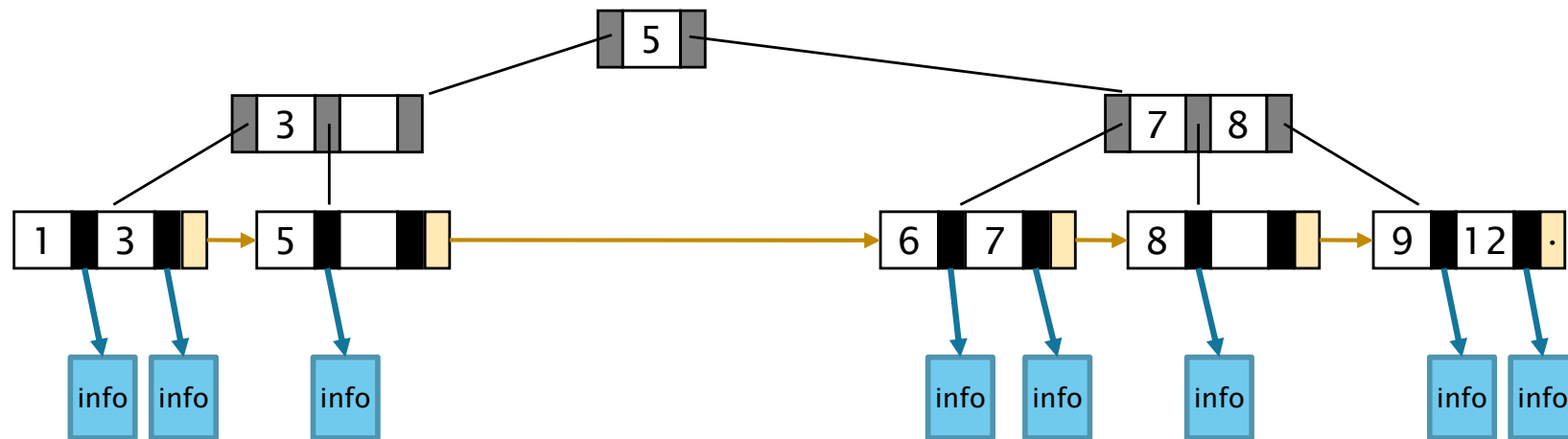
possuem apenas chaves

nós-folha

possuem chaves + dados

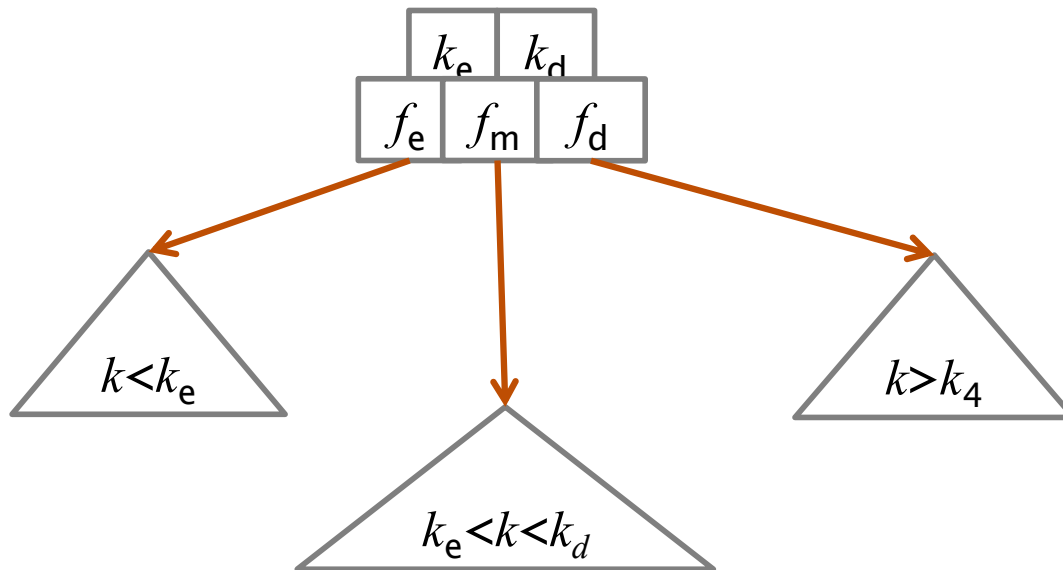
formam uma lista (duplamente) encadeada

Árvores B+ Exemplo

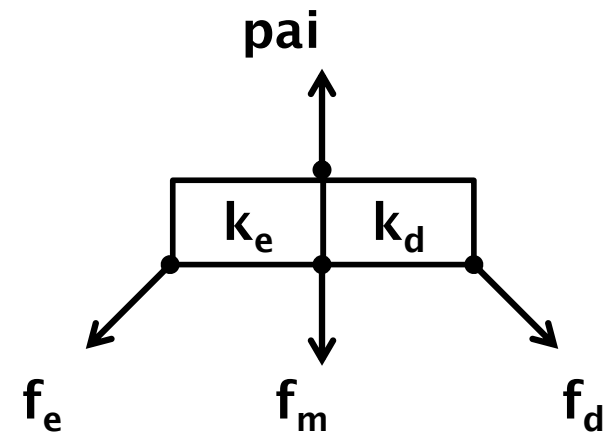


Árvore 2-3

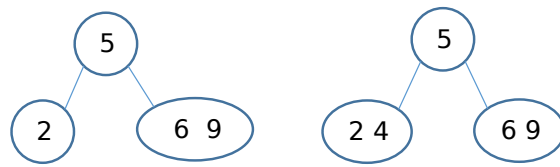
Árvore de busca com até 2 chaves e 3 filhos nos nós



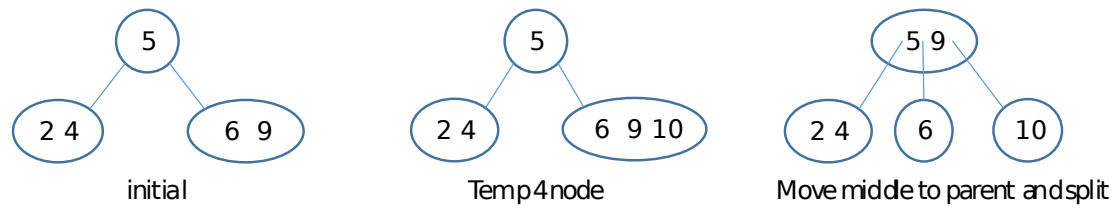
```
struct Btree23 {  
    int ke, kd;  
    Btree23 *pai;  
    Btree23 *fe, *fm, *fr;  
};
```



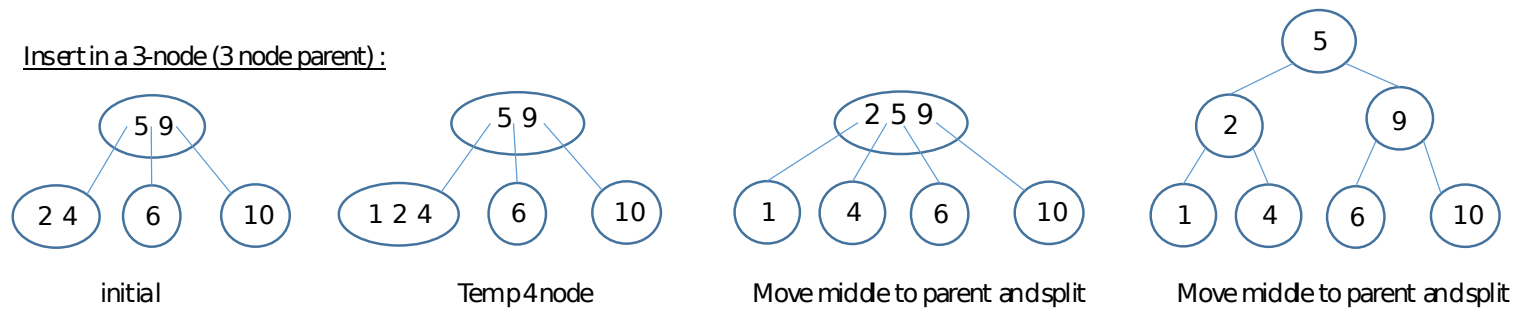
Insert in a 2-node :



Insert in a 3-node (2 node parent) :



Insert in a 3-node (3 node parent) :



Diaa abdelmoneim