

Aula 27: Algoritmos em Árvores B

- ⇒ Busca de uma chave
- ⇒ Inserção de uma chave
- ⇒ Remoção de uma chave

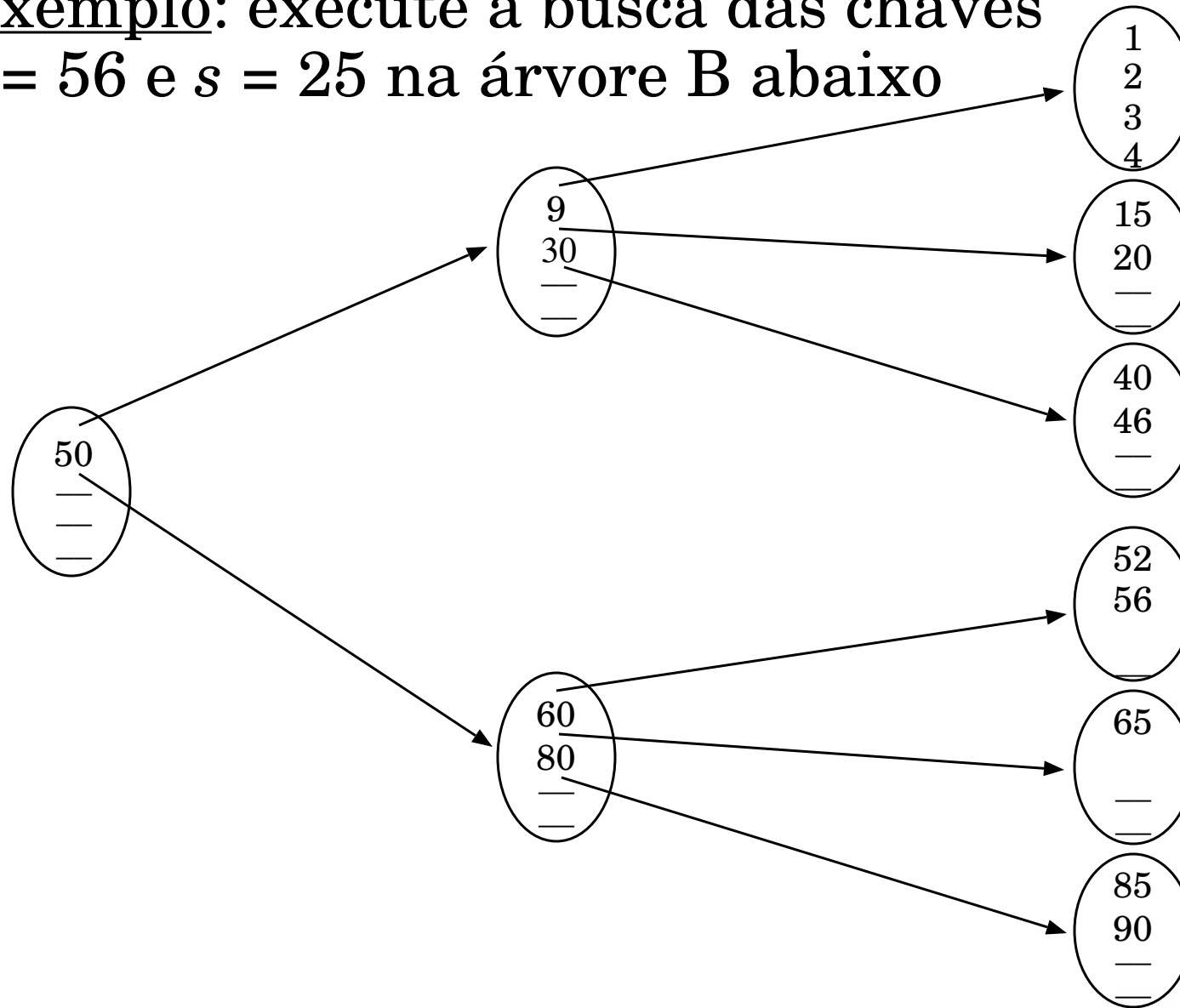
Busca de uma chave em uma árvore B

- ➡ Seja s a chave procurada
- ➡ Inicialmente, percorremos a lista de chaves ordenadas na página-raiz
- ➡ Se s for encontrada, o processo termina
- ➡ Caso contrário, continua-se a busca na página-filha apontada pelo ponteiro adequado
 - ▢ Se $s < s_1$, usar p_0
 - ▢ Se $s_i < s < s_{i+1}$, usar p_i ($1 \leq i \leq m-1$)
 - ▢ Se $s > s_m$, usar p_m
- ➡ Repete-se a busca na página-filha, e assim por diante, até que:
 - ▢ ou s seja encontrada
 - ▢ ou atinja-se um ponteiro nulo

Busca de uma chave em uma árvore B



Exemplo: execute a busca das chaves $s = 56$ e $s = 25$ na árvore B abaixo

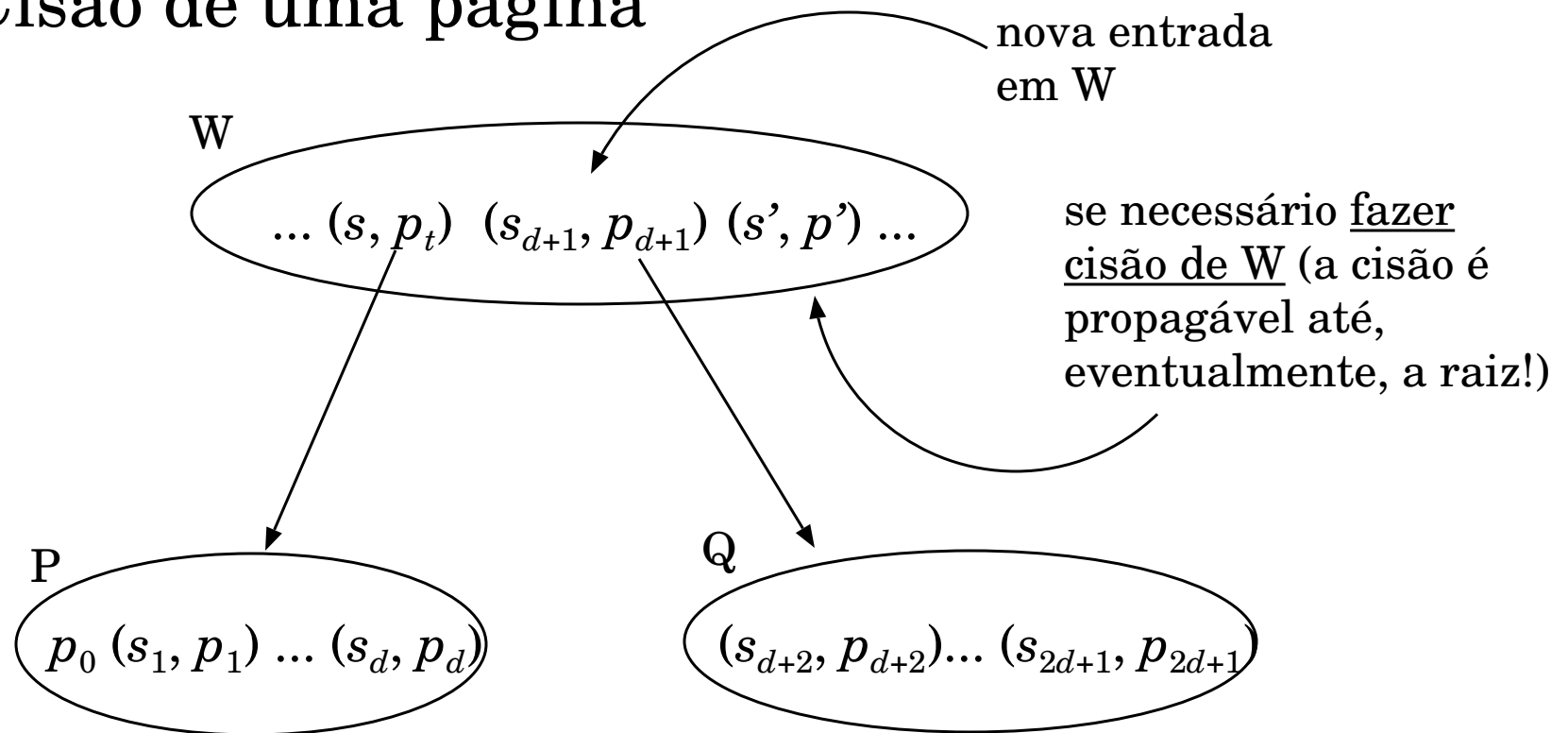


Inserção de uma chave em uma árvore B

- ➡ Seja s a chave a ser inserida
- ➡ Inicialmente, fazemos a busca da chave s na árvore B
- ➡ Se s for encontrada, nada a fazer (pois nesse caso a inserção seria inválida)
- ➡ Se s não for encontrada, insere-se s na página-folha onde a busca se encerrou, mantendo a ordenação correta da lista de chaves
- ➡ Se eventualmente a página-folha contém mais de $2d$ chaves, é preciso efetuar sua **cisão** (lembre-se que $2d$ é o limite máximo para o número de chaves em uma página)

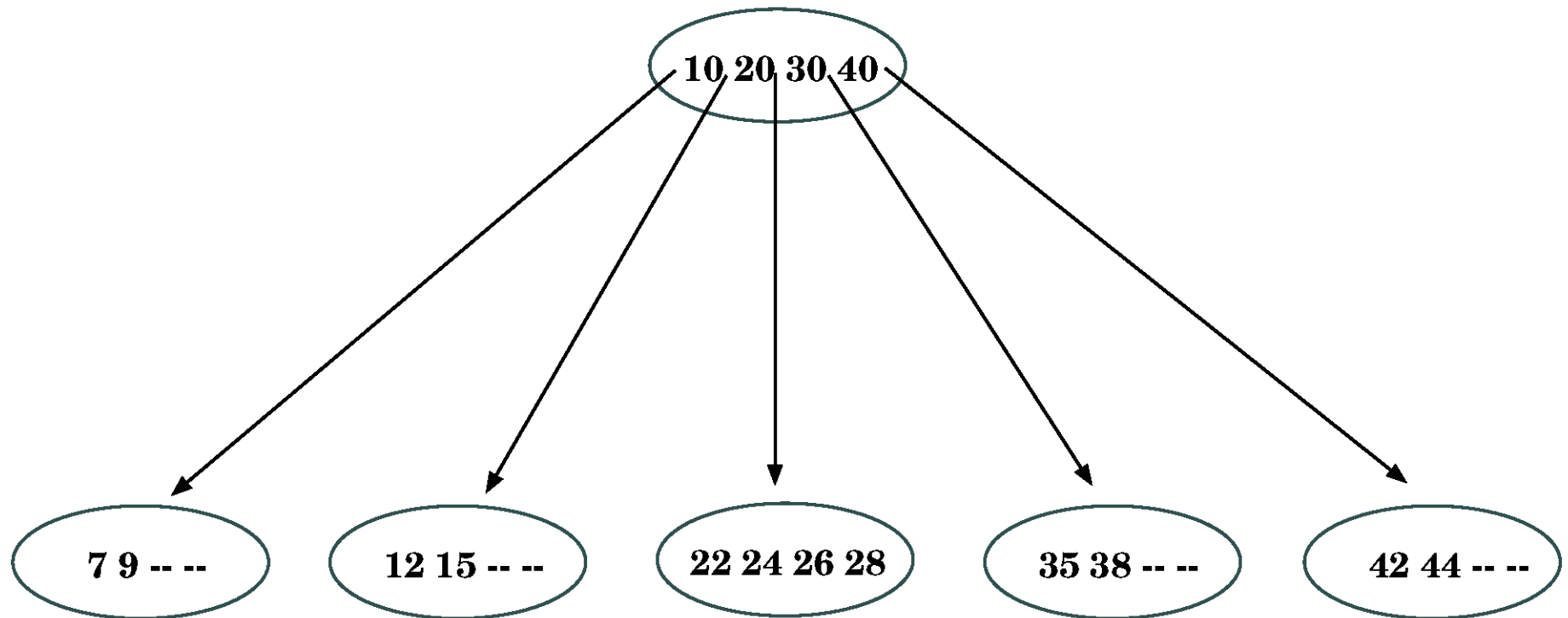
Inserção de uma chave em uma árvore B

⇒ Cisão de uma página



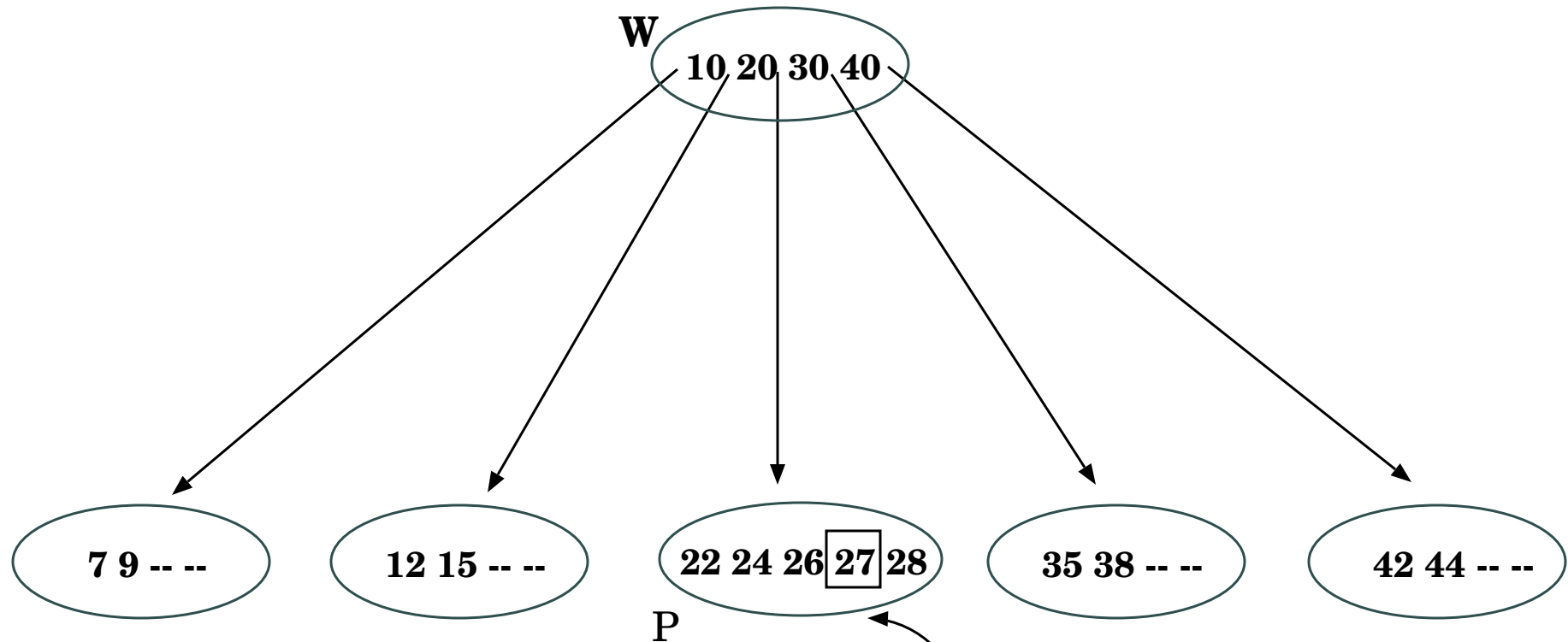
Inserção de uma chave em uma árvore B

➡ Exemplo: inserir a chave $s = 27$ na árvore B
de ordem 2 abaixo



Inserção de uma chave em uma árvore B

➡ Exemplo (continuação): Após a busca de $s = 27$, a inserção é realizada na página P

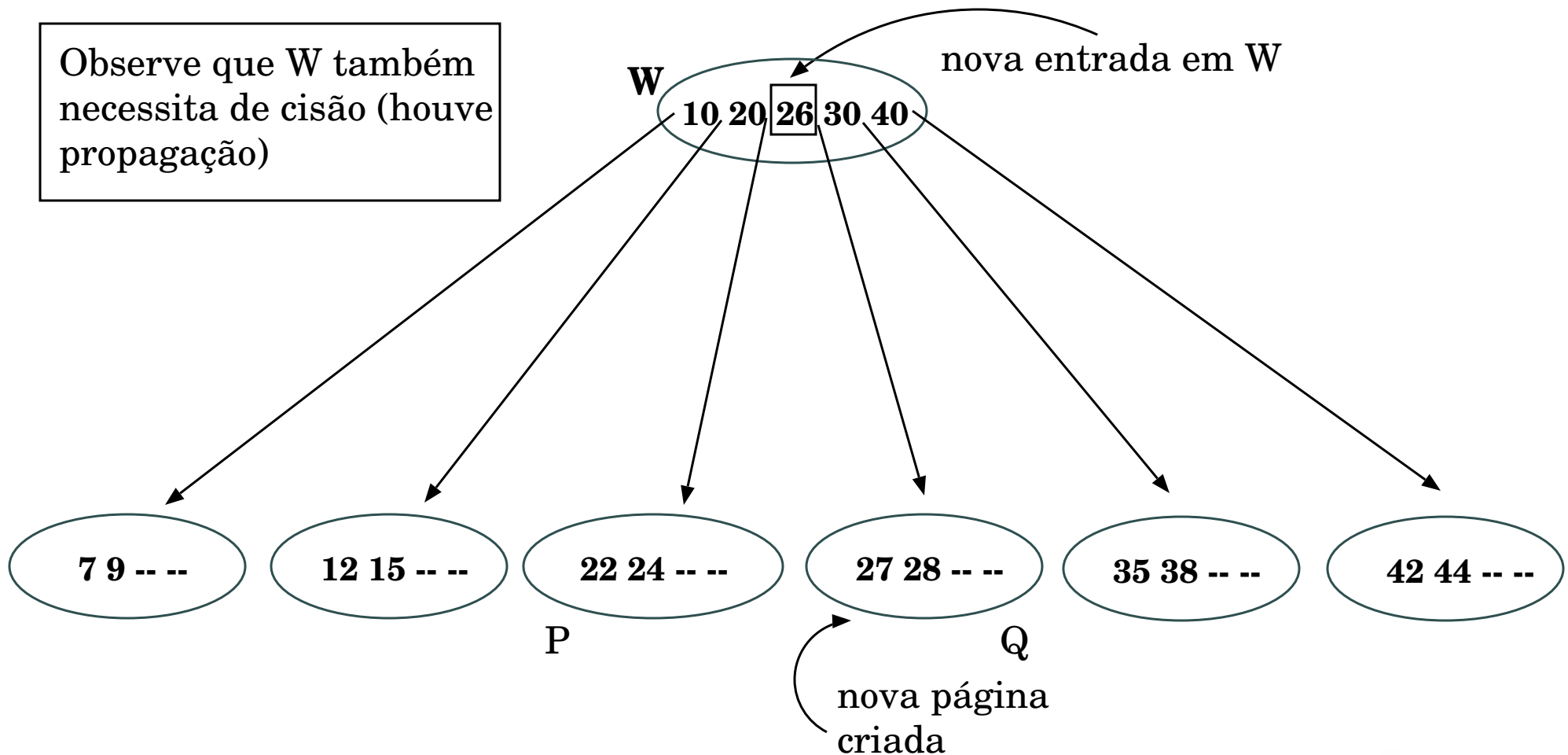


Como contém $2 \times 2 + 1 = 5$ chaves, necessita cisão

Inserção de uma chave em uma árvore B

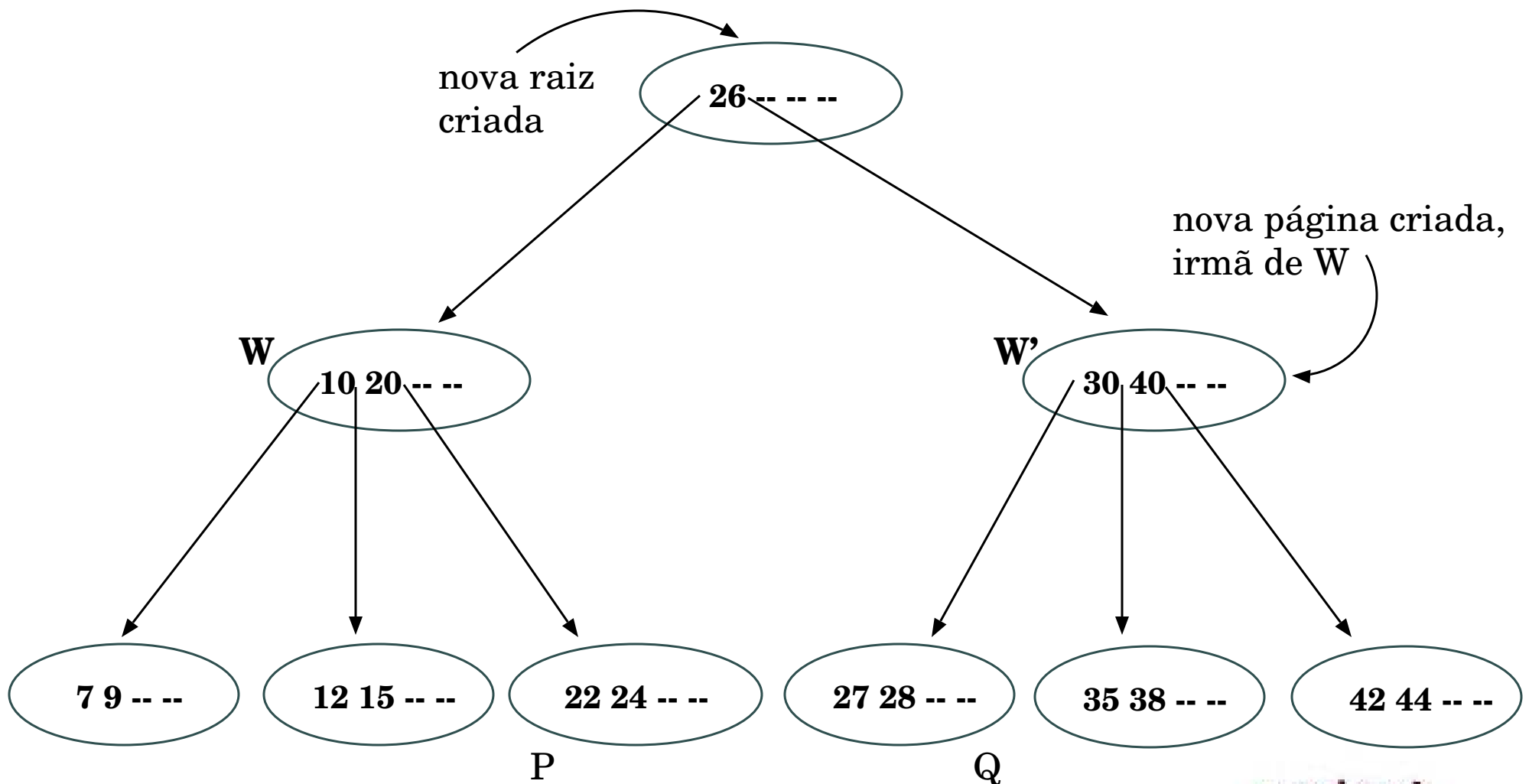
➡ Exemplo (continuação): Efetuamos a cisão de P

Observe que W também
necessita de cisão (houve
propagação)



Inserção de uma chave em uma árvore B

➡ Exemplo (continuação): Efetuamos a cisão de W



Remoção de uma chave em uma árvore B

- ➡ Seja s a chave a ser removida
- ➡ Inicialmente, fazemos a busca da chave s na árvore B
- ➡ Se s não for encontrada, nada a fazer (pois nesse caso a remoção seria inválida)
- ➡ Se s for encontrada numa folha, remova-a
- ➡ Se s for encontrada numa página não-folha, remova-a e coloque no seu lugar a chave x imediatamente maior que s (x sempre se encontra numa folha)
- ➡ A remoção se restringe portanto sempre a uma página folha

Remoção de uma chave em uma árvore B

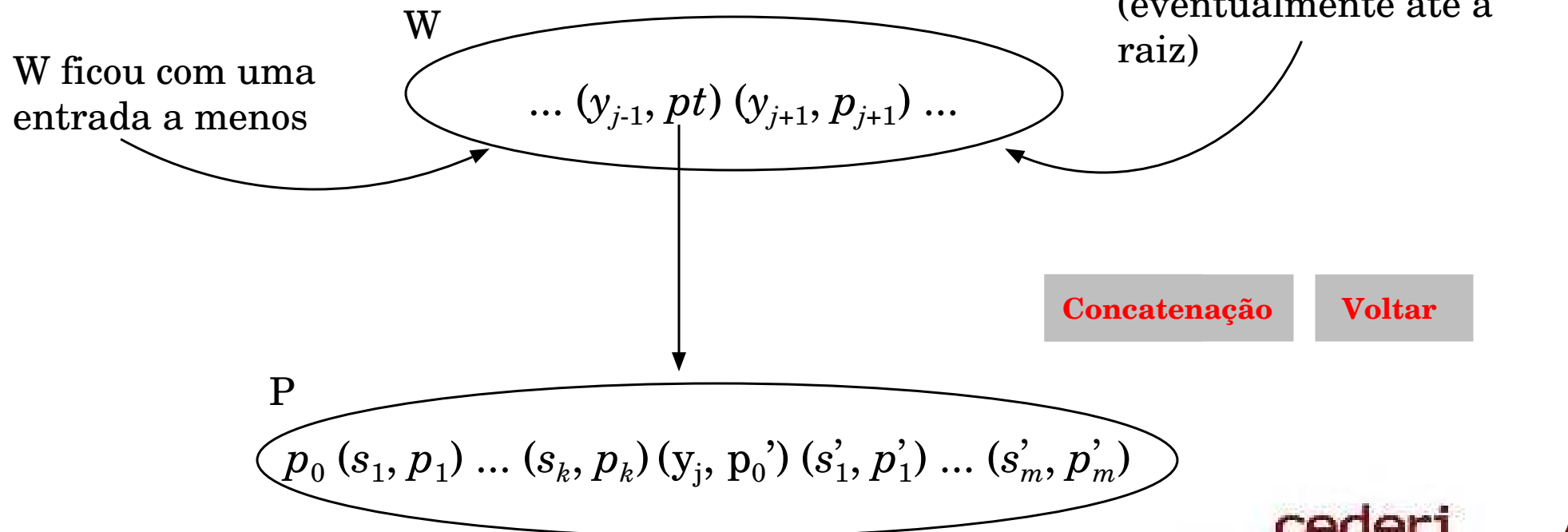
- ➡ Se a página-folha P onde foi realizada a remoção ficou com menos do que d chaves, é preciso fazer **concatenação** ou **redistribuição**
- ➡ Vejamos em que casos aplicamos **concatenação** ou **redistribuição**

Remoção de uma chave em uma árvore B

⇒ Concatenação

- Aplicamos concatenação quando a página P (com $k < d$ chaves) possui uma irmã Q (com m chaves) tal que

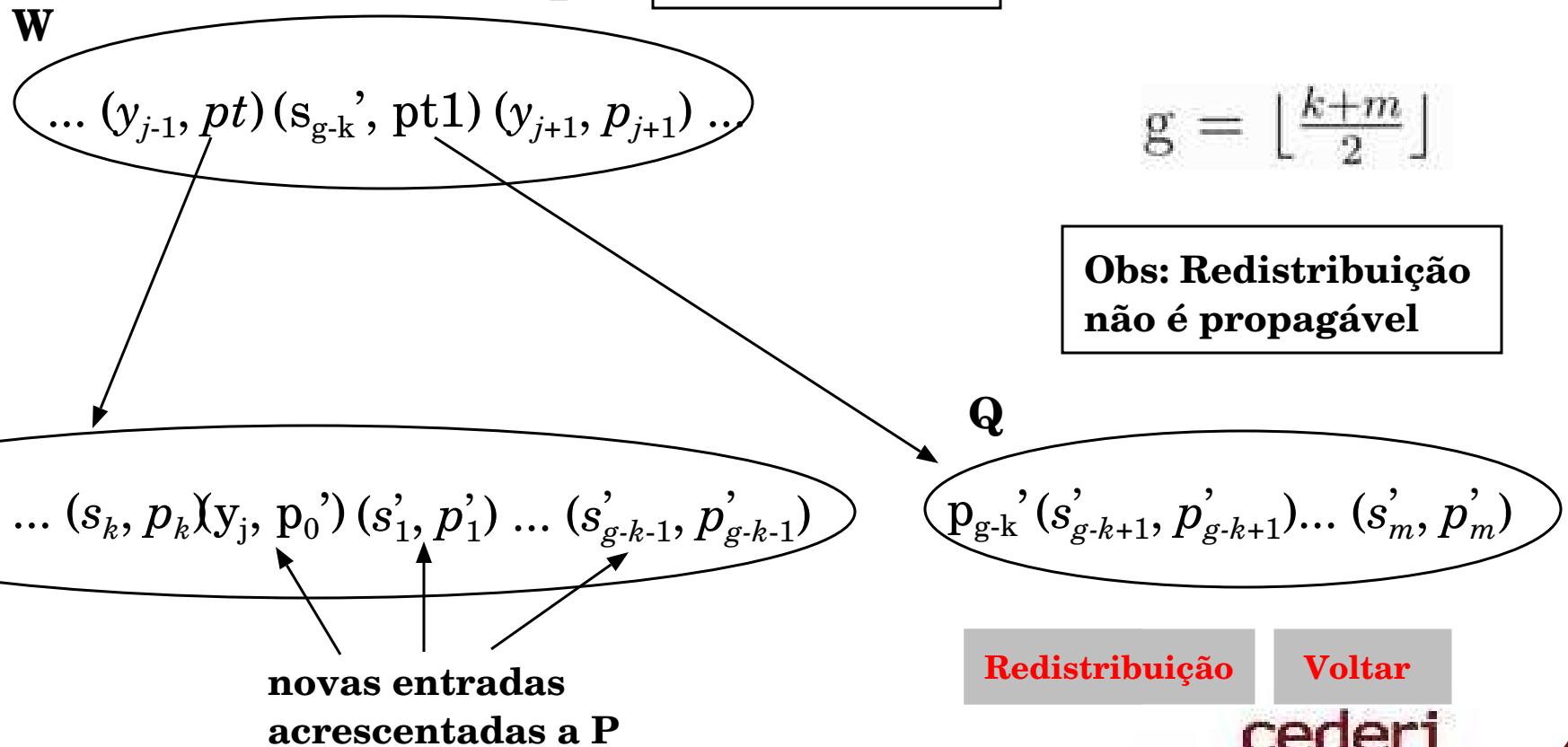
$$k + m < 2d$$



Remoção de uma chave em uma árvore B

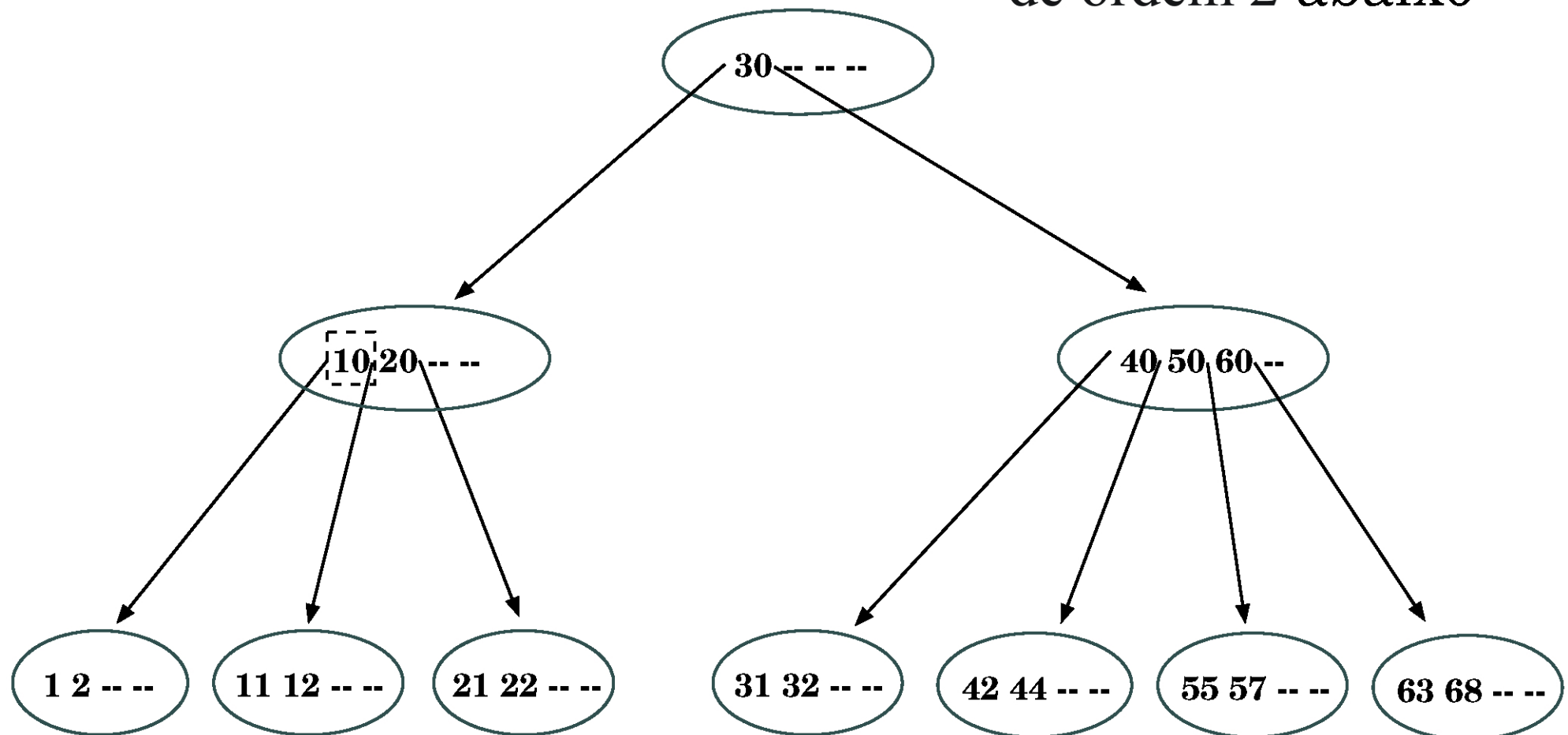
⇒ Redistribuição

- ▬ Aplicamos redistribuição quando a página P (com $k < d$ chaves) possui uma irmã Q (com m chaves) tal que $k + m \geq 2d$



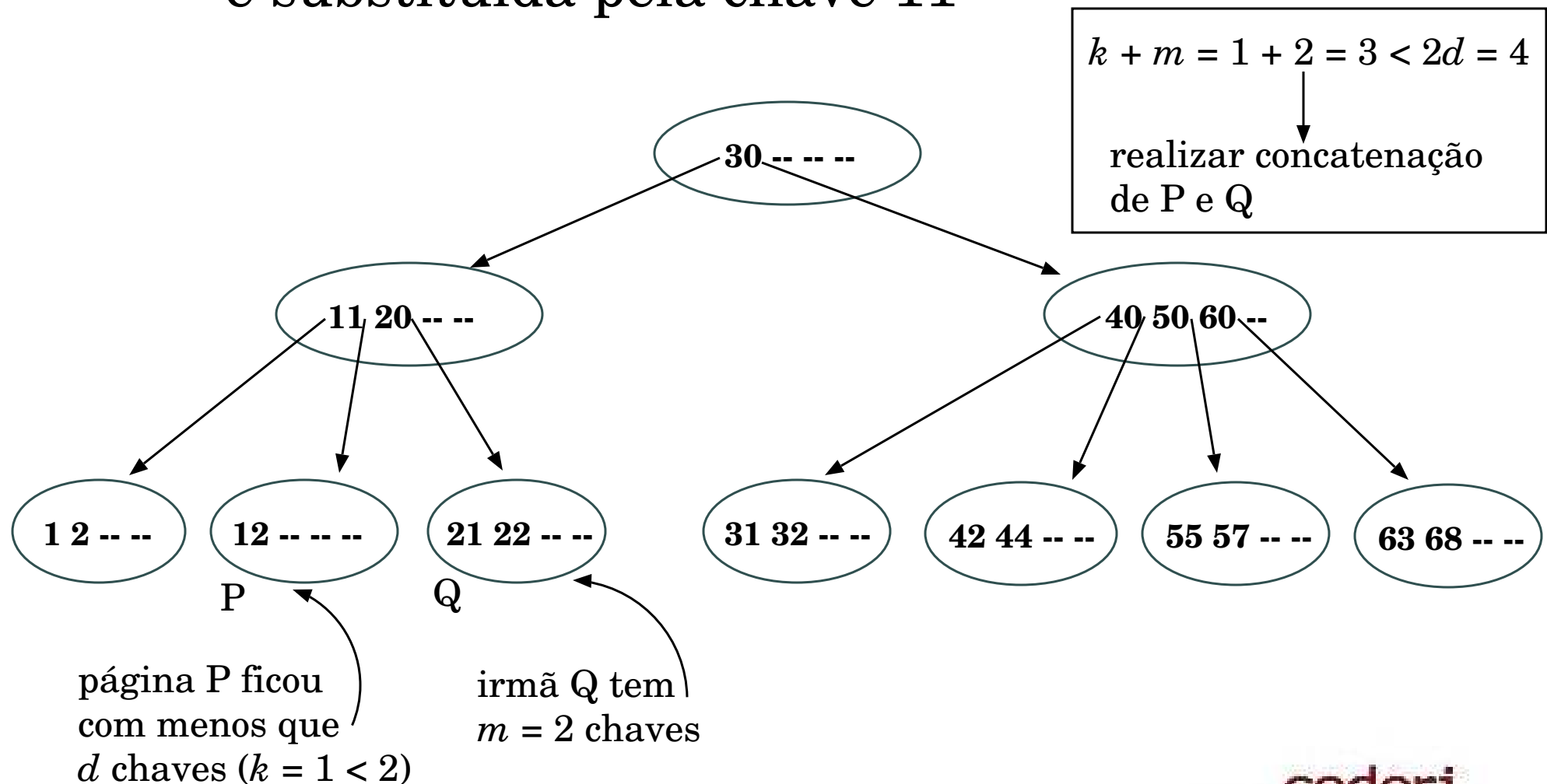
Remoção de uma chave em uma árvore B

⇒ Exemplo: Remover a chave $s = 10$ da árvore B
de ordem 2 abaixo



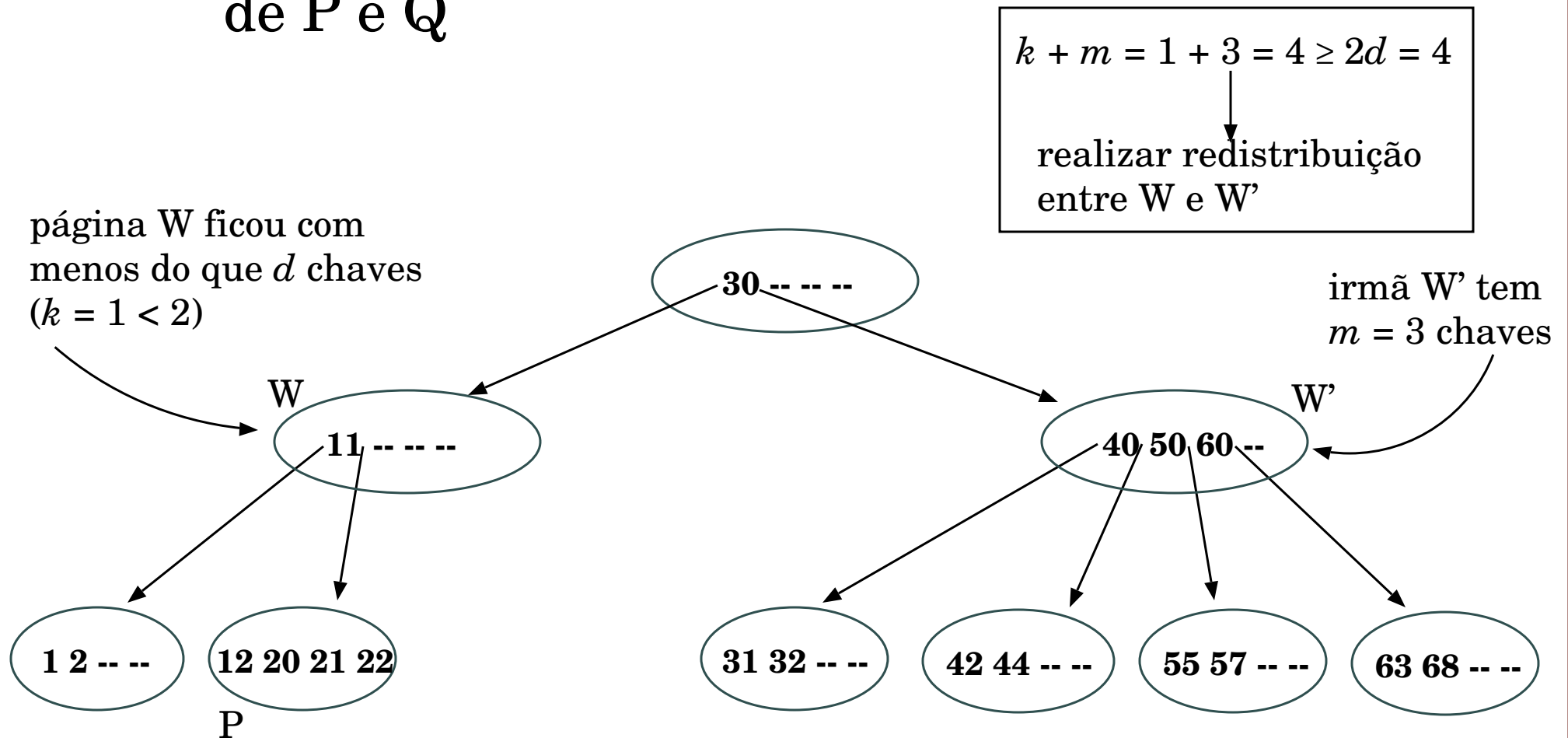
Remoção de uma chave em uma árvore B

➡ Exemplo (continuação): a chave é 10 removida e substituída pela chave 11



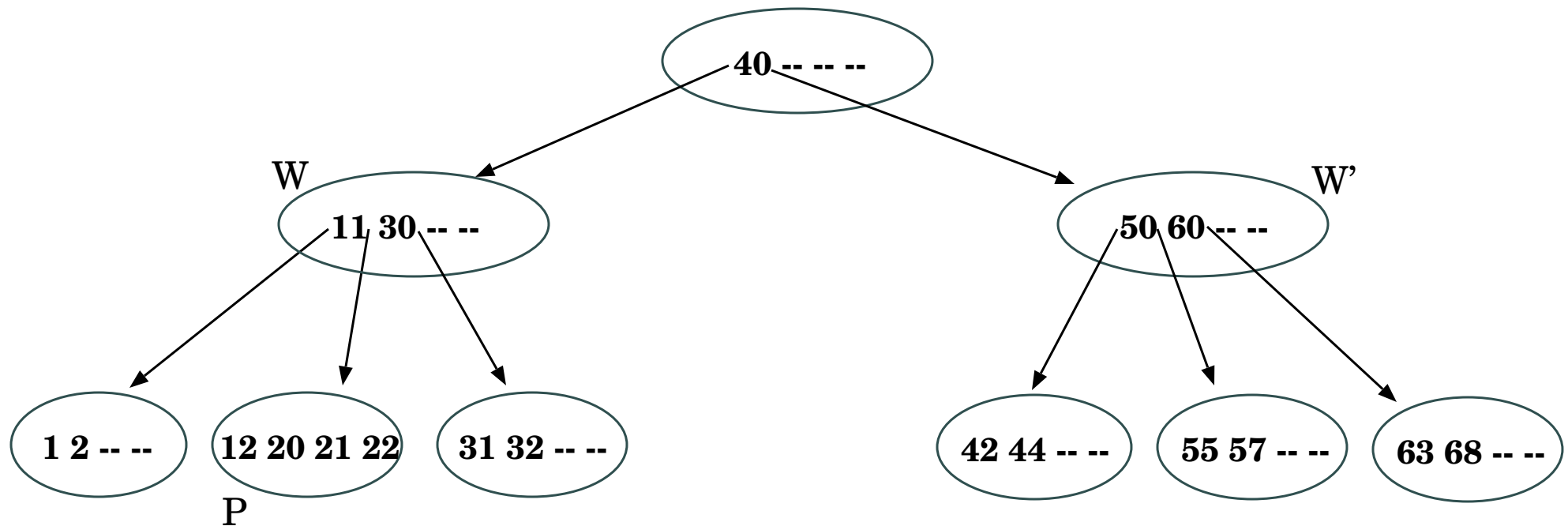
Remoção de uma chave em uma árvore B

➡ Exemplo (continuação): realizamos a concatenação de P e Q



Remoção de uma chave em uma árvore B

⇒ Exemplo (continuação): realizamos a redistribuição entre W e W'



Exercício final

- ➡ Desenhe uma árvore B de ordem 2 e altura 3 e indique uma chave cuja remoção exija uma concatenação propagável até a raiz
- ➡ Desenhe a árvore resultante após as concatenações
(observe que a árvore resultante terá altura diminuída em uma unidade!)