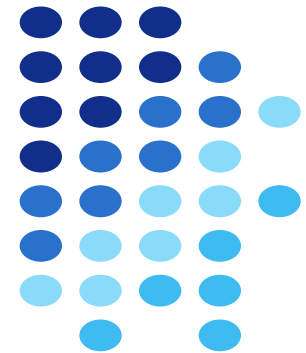


Universidade Federal de Sergipe
Departamento de Sistemas de Informação
SINF0007 – Estrutura de Dados II

Árvore B

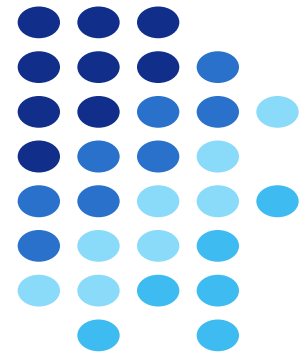


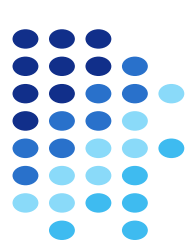
6

Prof. Dr. Raphael Pereira de Oliveira

raphael.oliveira@academico.ufs.br

Introdução



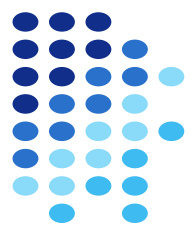


Consulta a Arquivos Binários Grandes

- Arquivos binários grandes
 - Busca sequencial é muito custosa
 - Se arquivo estiver ordenado pode-se fazer busca binária, mas para arquivos grandes ainda não é eficiente o suficiente
- É possível acelerar a busca usando duas técnicas:
 - Acesso via cálculo do endereço do registro (*hashing*)
 - Acesso via estrutura de dados auxiliar (índice)

Índice

- Índice é uma estrutura de dados que serve para localizar registros no arquivo de dados
- Cada entrada do índice contém
 - Valor da chave
 - Ponteiro para o arquivo de dados
- Pode-se pensar então em dois arquivos:
 - Um de índice
 - Um de dados
- **Isso é eficiente?**



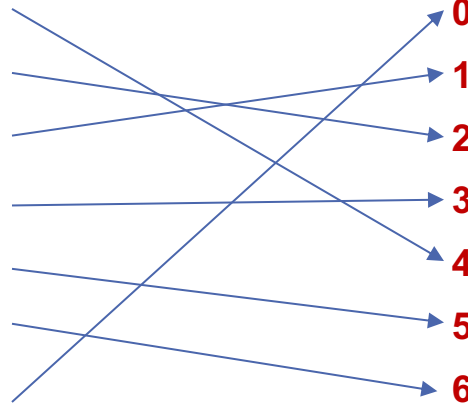
Exemplo de Índice Plano

Arquivo de Índice

	<u>CHAVE</u>	<u>PONTEIRO</u>
0	3	4
1	5	2
2	10	1
3	15	3
4	16	5
5	21	6
6	23	0

Arquivo de Dados

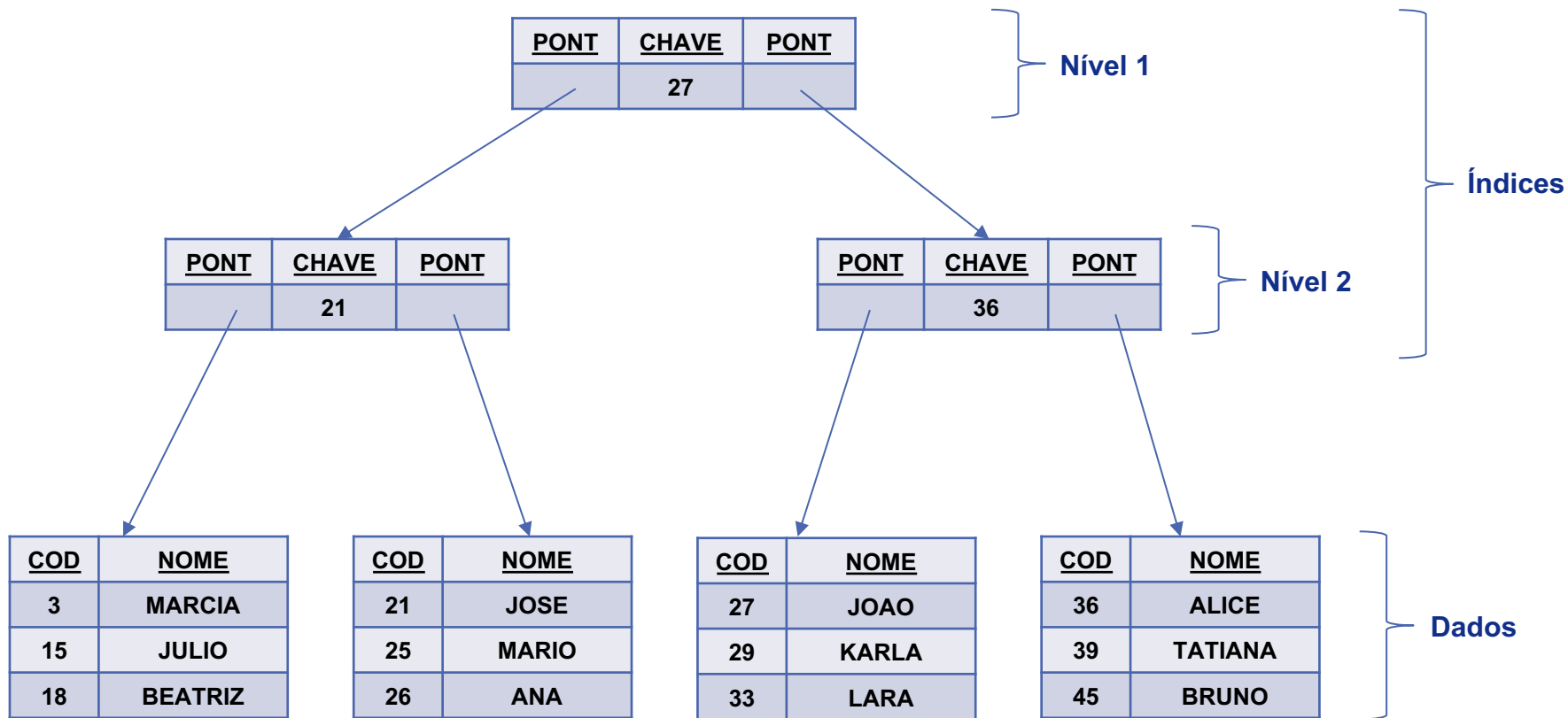
<u>COD</u>	<u>NOME</u>
23	JOSE
10	MARIO
5	ANA
15	MARCIA
3	JULIO
16	BEATRIZ
21	CAMILA

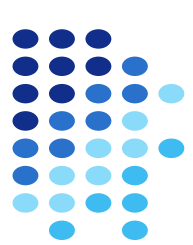


Índice

- Se tivermos que percorrer o arquivo de índice sequencialmente para encontrar uma determinada chave, o índice não terá muita utilidade
 - Pode-se fazer busca um pouco mais eficiente (ex. busca binária), se o arquivo de índice estiver ordenado
 - Mas mesmo assim isso não é o ideal
- Para resolver este problema:
 - os índices não são estruturas sequenciais, e sim hierárquicas
 - os índices não apontam para um registro específico, mas para um bloco de registros (e dentro do bloco é feita busca sequencial) – exige que os registros dentro de um bloco estejam ordenados

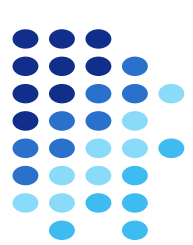
Exemplo de Índice Hierárquico





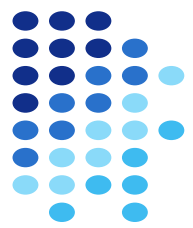
Hierarquia Lembra árvore...

- A maioria das estruturas de índice é implementada por árvores de busca
 - Árvores de Busca Binária
 - Árvores AVL
 - Árvores de Múltiplos Caminhos

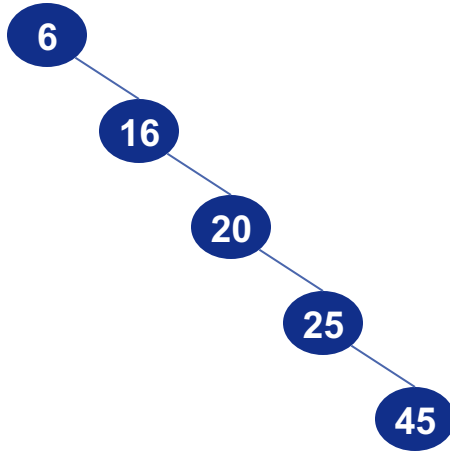


Árvore de Busca Binária

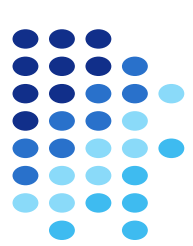
- **Relembrando:** Características de uma árvore de busca binária T
 - todas as chaves da subárvore da esquerda de T têm valores menores que a chave do nó raiz de T
 - todas as chaves da subárvore da direita de T têm valores maiores que a chave do nó raiz de T
 - as subárvores esquerda e direita de T também são árvores de busca binária



Considerações sobre Árvores de Busca Binárias

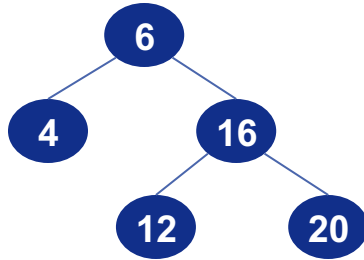


- Altura tende a ser muito grande em relação ao número de nós ou registros que ela contém
- Se as chaves a serem incluídas estiverem ordenadas, a árvore degrada-se rapidamente, tornando-se uma lista encadeada

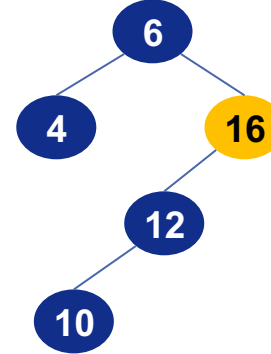


Árvores AVL

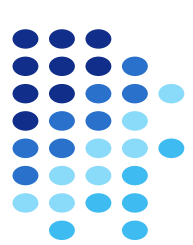
- São árvores binárias balanceadas
 - Para qualquer nó da árvore, a altura da subárvore da esquerda não pode diferir em mais de 1 unidade da altura da subárvore da direita



Árvore AVL

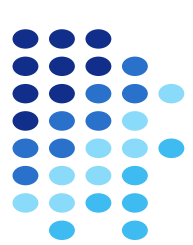


Árvore Não AVL



Considerações sobre Árvores AVL

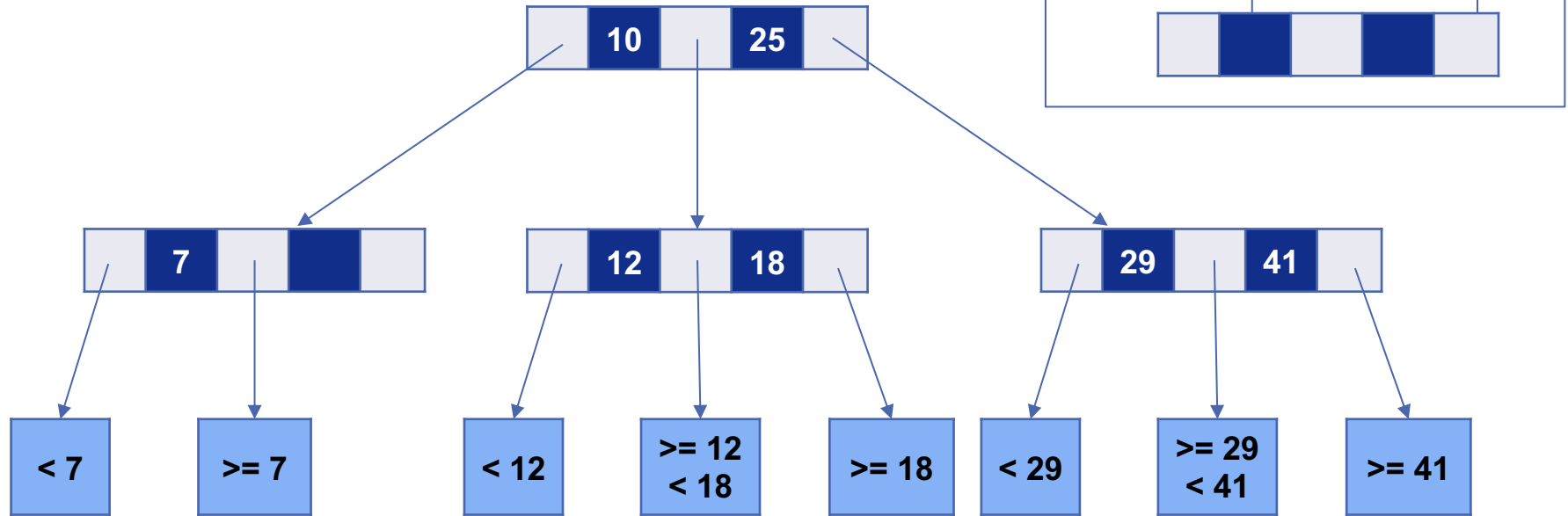
- Ainda são excessivamente altas para uso eficiente como estrutura de índice

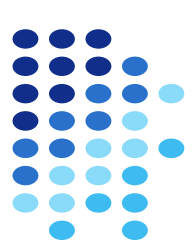


Solução: Árvores de Múltiplos Caminhos

- Características
 - Cada nó contém **$n-1$** chaves
 - Cada nó contém **n** filhos
 - As chaves dentro do nó estão ordenadas
 - As chaves dentro do nó funcionam como separadores para os ponteiros para os filhos do nó

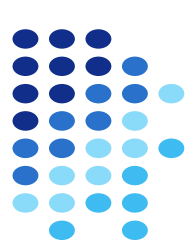
Exemplo





Vantagens

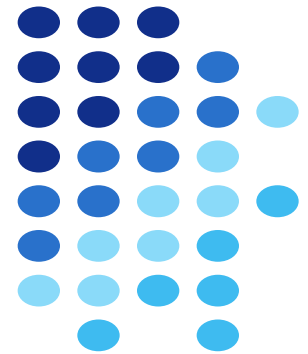
- Têm altura bem menor que as árvores binárias
- Ideais para uso como índice de arquivos em disco
- Como as árvores são baixas, são necessários poucos acessos em disco até chegar ao ponteiro para o bloco que contém o registro desejado

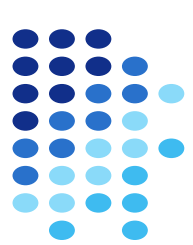


Exemplos de Árvores de Múltiplos Caminhos

- Árvore B
- Árvore B+

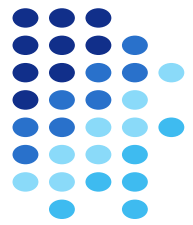
Árvore B





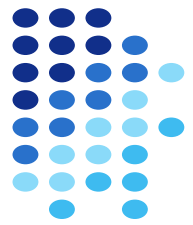
Árvore B

- Consegue armazenar índice e dados na mesma estrutura (mesmo arquivo físico)
- Características de uma **árvore B** de ordem **d**
 - A raiz é uma folha ou tem no mínimo 2 filhos
 - Cada nó interno (não folha e não raiz) possui no mínimo **d + 1** filhos
 - Cada nó tem no máximo **2d + 1** filhos
 - Todas as folhas estão no mesmo nível
- Um nó de uma **árvore B** é também chamado de página
- Uma página armazena diversos registros da tabela original
 - Seu tamanho normalmente equivale ao tamanho de uma página em disco



Árvore B

- Outras propriedades
 - Seja m o número de chaves de uma página P não folha
 - P tem $m + 1$ filhos, P tem entre d e $2d$ chaves, exceto o nó raiz, que possui entre 1 e $2d$ chaves
 - Em cada página, as chaves estão ordenadas: s_1, \dots, s_m , onde $d \leq m \leq 2d$, exceto para a raiz onde $1 \leq m \leq 2d$
 - P contém $m + 1$ ponteiros p_0, p_1, \dots, p_m para os filhos de P
 - Nas páginas correspondentes às folhas, esses ponteiros apontam para **NULL**
 - Os nós também armazenam, além da chave s_k , os dados (I_k) relativos àquela chave



Árvore B

- Seja uma página **P** com **m** chaves:
 - para qualquer chave **y** pertencente à primeira página apontada por **P** (ou seja, apontada por **p₀**), **y < s₁**
 - para qualquer chave **y** pertencente à página apontada por **p_k**, **1 ≤ k ≤ m-1**, **s_k < y < s_{k+1}**
 - para qualquer chave **y** pertencente à página apontada por **p_m**, **y > s_m**

Estrutura de uma página (nó)



Características de uma **árvore B** de ordem **d**

A raiz é uma folha ou tem no mínimo 2 filhos

Cada nó tem entre **d** e **2d** chaves, exceto o nó raiz, que possui entre **1** e **2d** chaves

Cada nó interno (não folha e não raiz) possui no mínimo **d + 1** filhos

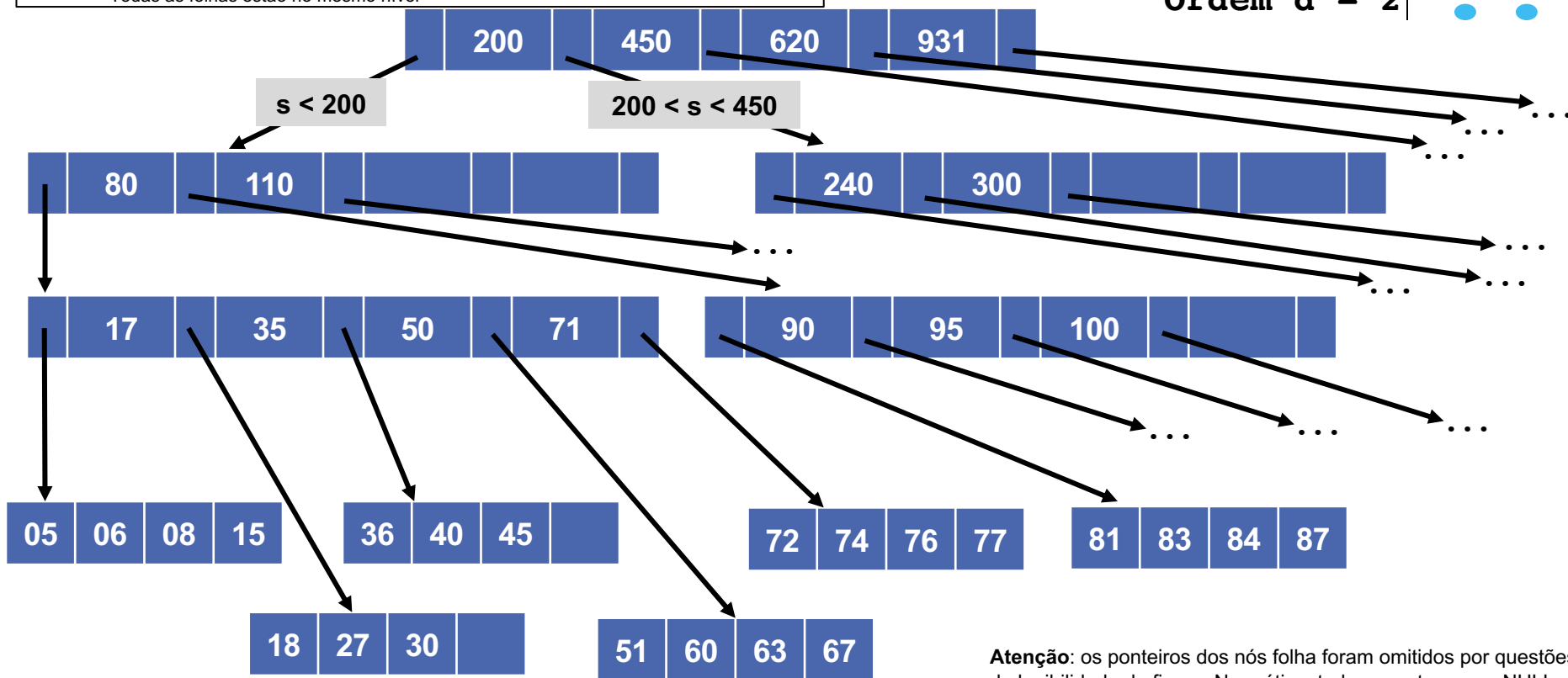
Cada nó tem no máximo **2d + 1** filhos

Todas as folhas estão no mesmo nível

Exemplo

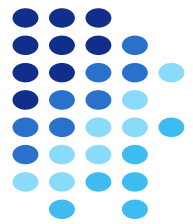


Ordem **d = 2**



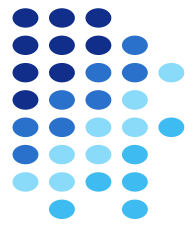
Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

Figura: Vanessa Braganholo (slides de árvore B)



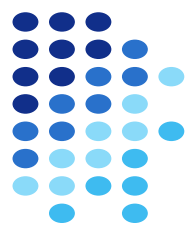
Representação em C

```
typedef struct No {  
    int m; //quantidade de chaves armazenadas no nó  
    struct No *pont_pai; //pt para o nó pai  
    int *s; //array de chaves  
    struct No **p; //pt para array de pt p/ os filhos  
} TNo;
```



Busca de uma Chave X em uma Árvore B que Indexa Arquivo em Disco

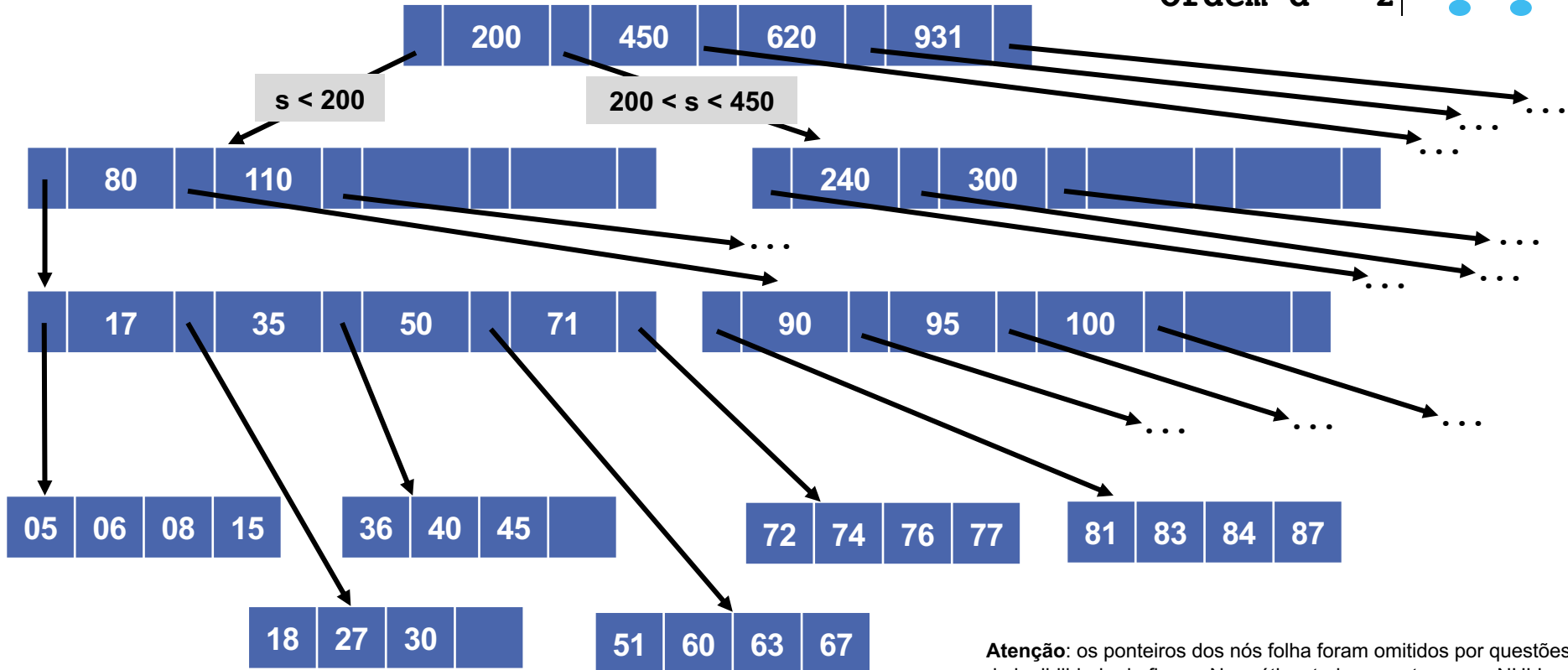
1. Inicie lendo a raiz da árvore a partir do disco
2. Procure **x** dentro do nó lido (pode ser usada busca binária, pois as chaves estão ordenadas dentro do nó)
 - a) Se encontrou, encerra a busca;
 - b) Caso contrário, continue a busca, lendo o filho correspondente, a partir do disco
3. Continue a busca até que **x** tenha sido encontrado ou que a busca tenha sido feita em uma folha da árvore (retorna o último nó pesquisado – nó onde a chave está ou deveria estar)



Exemplo

Buscar chaves 240, 76 e 85 na árvore

Ordem $d = 2$



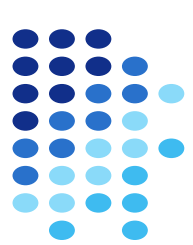
Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

Figura: Vanessa Braganholo (slides de árvore B)



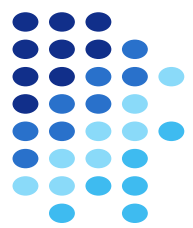
Função de Busca

```
TNo *busca(TNo *no, int ch) {  
    if (no != NULL) {  
        int i = 0;  
        while (i < no->m && ch > no->s[i]) {  
            i++;  
        }  
        if (i < no->m && ch == no->s[i]) {  
            return no; // encontrou chave  
        } else if (no->p[i] != NULL) {  
            return busca(no->p[i], ch);  
        } else return no; //nó era folha -- não existem mais  
                                //nós a buscar, então retorna o nó onde a chave  
                                //deveria estar  
    } else return NULL; //nó é NULL, não há como buscar  
}
```



Inserção

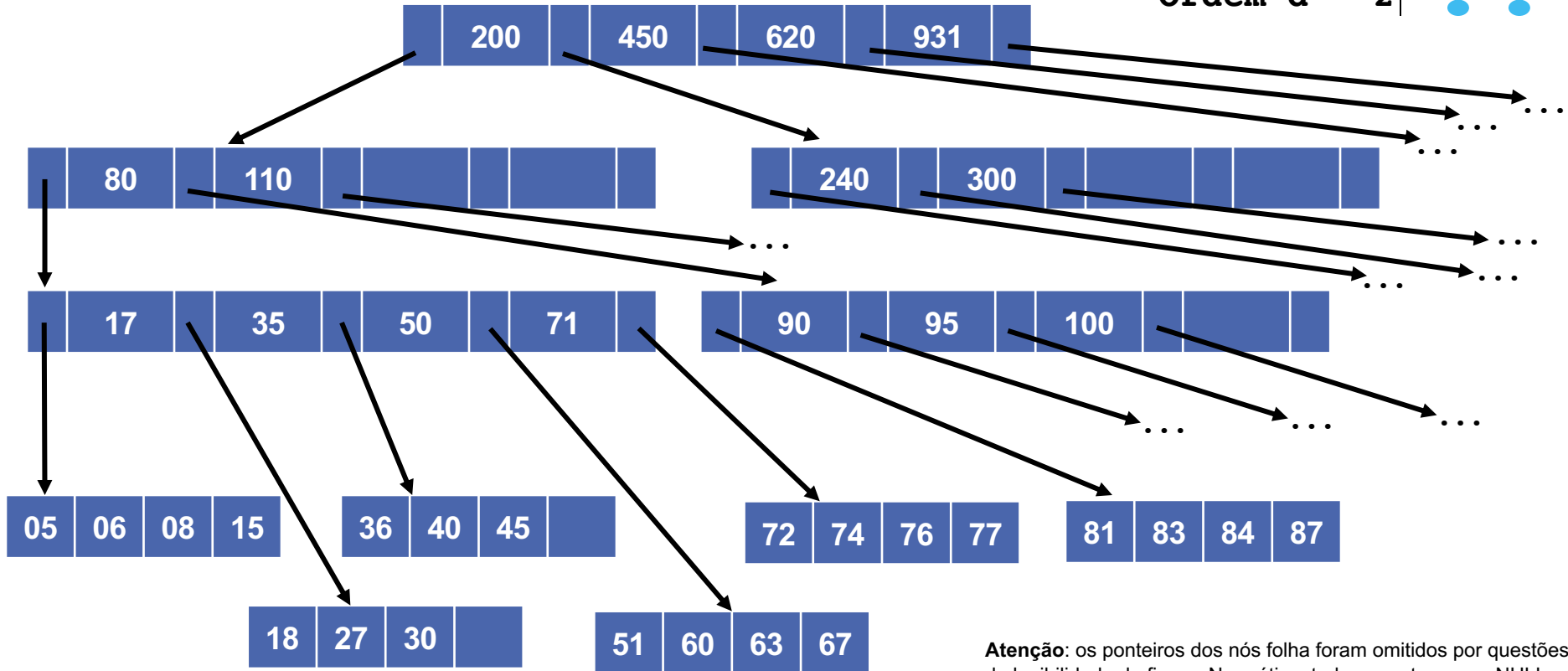
- Para inserir um registro de chave **x** na **árvore B**
 - Executar o algoritmo de busca
 - Se chave está no nó retornado pela busca (é preciso checar)
 - Inserção é inválida
 - Se chave não está no nó retornado pela busca:
 - Inserir a chave no nó retornado pela busca



Inserção

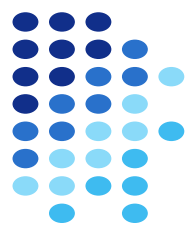
Inserir chave 32

Ordem $d = 2$



Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

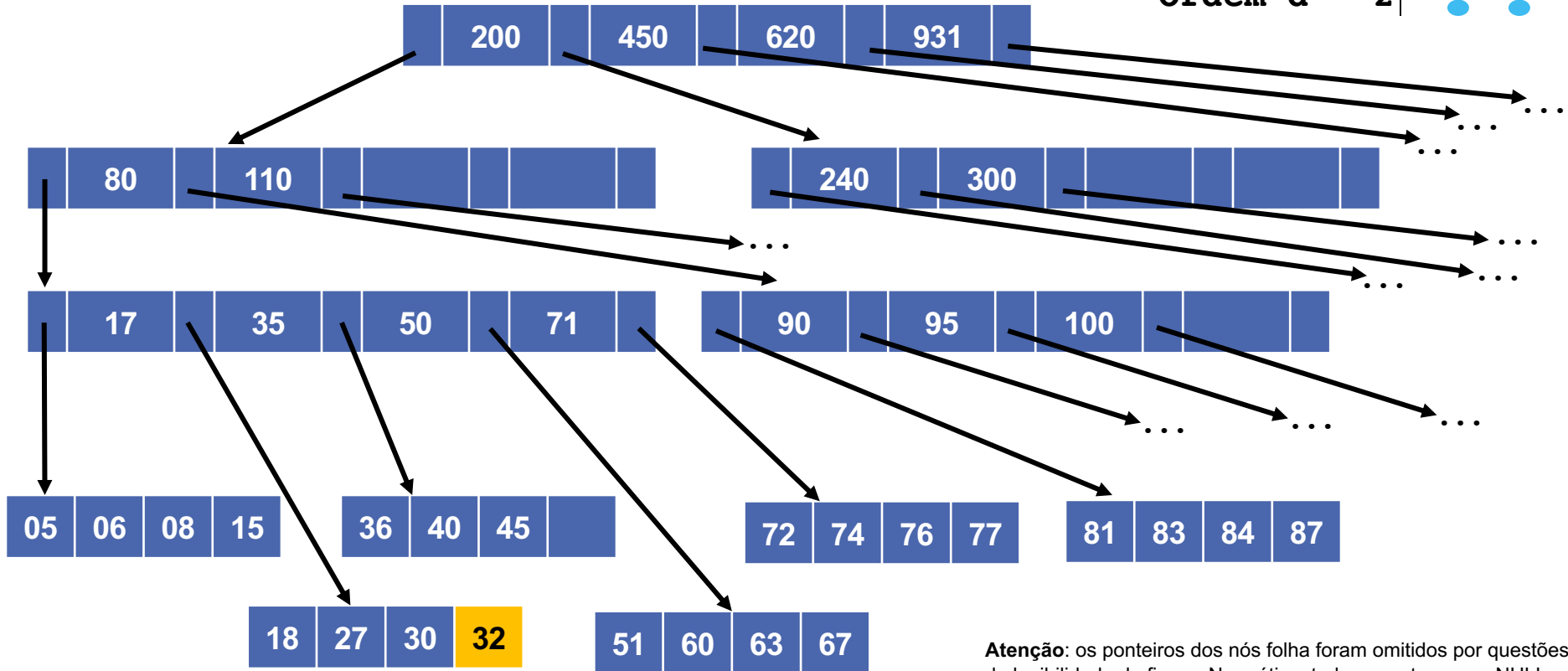
Figura: Vanessa Braganholo (slides de árvore B)



Inserção

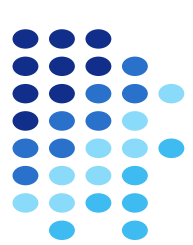
Inserir chave 32

Ordem $d = 2$



Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

Figura: Vanessa Braganholo (slides de árvore B)



Inserção

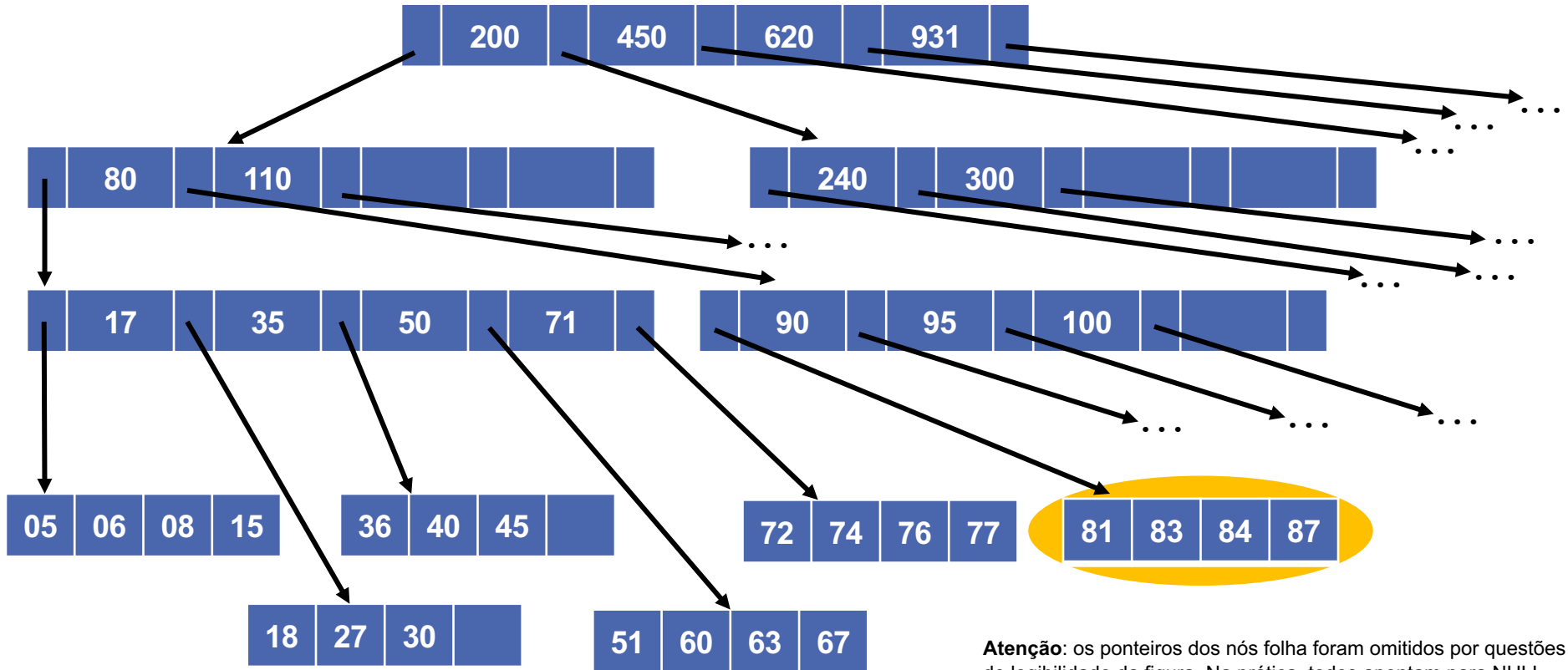
Inserção ocorre sempre nas
folhas!

Inserção

Inserir chave 85

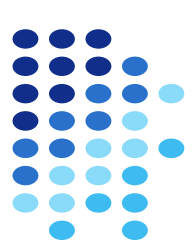
Inserção faria a página ficar com $2d + 1$ chaves

Ordem $d = 2$



Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

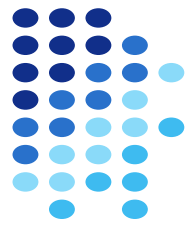
Figura: Vanessa Braganholo (slides de árvore B)



Problema: Página Cheia

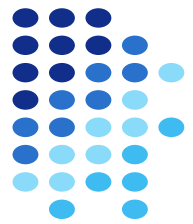
- É necessário reorganizar as páginas
- Ao inserir uma chave em uma página cheia, sua estrutura ficaria da seguinte forma (omitindo l_k para simplificar)
 - $p_0, (s_1, p_1), (s_2, p_2), \dots, (s_d, p_d), (s_{d+1}, p_{d+1}), \dots, (s_{2d+1}, p_{2d+1})$





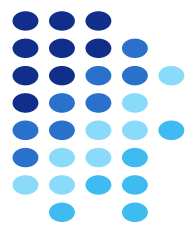
Solução

- Particionar a página em 2
 - Na página **P** permanecem **d** entradas
 - Entrada **d+1** sobe para o pai
 - Alocar outra página, **Q**, e nela alocar as outras **d** entradas
- Após o particionamento
 - Estrutura da página **P**:
 - $p_0, (s_1, p_1), (s_2, p_2), \dots, (s_d, p_d)$
 - Estrutura da página **Q**:
 - $p_{d+1}, (s_{d+2}, p_{d+2}) \dots, (s_{2d+1}, p_{2d+1})$



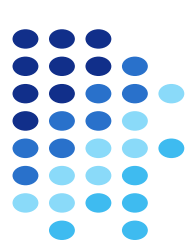
Alocação de S_{D+1}

- O nó **W**, agora também pai de **Q**, receberá a nova entrada (s_{d+1} , **pt**)
 - **pt** aponta para a nova página **Q**
- Se não houver mais espaço livre em **W**, o processo de particionamento também é aplicado a **W**



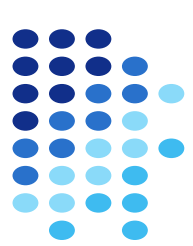
Particionamento

- **Observação importante:** particionamento se propaga para os pais dos nós, podendo, eventualmente, atingir a raiz da árvore
- O particionamento da raiz é a **única forma de aumentar a altura da árvore**



Procedimento de Inserção

1. Aplicar o procedimento busca, verificando a validade da inserção
2. Se a inserção é válida, realizar inserção no nó **F** retornado pela busca
3. Verificar se nó **F** precisa de particionamento. Se sim, propagar o particionamento enquanto for necessário



Exemplo de Inserção que Causa Particionamento

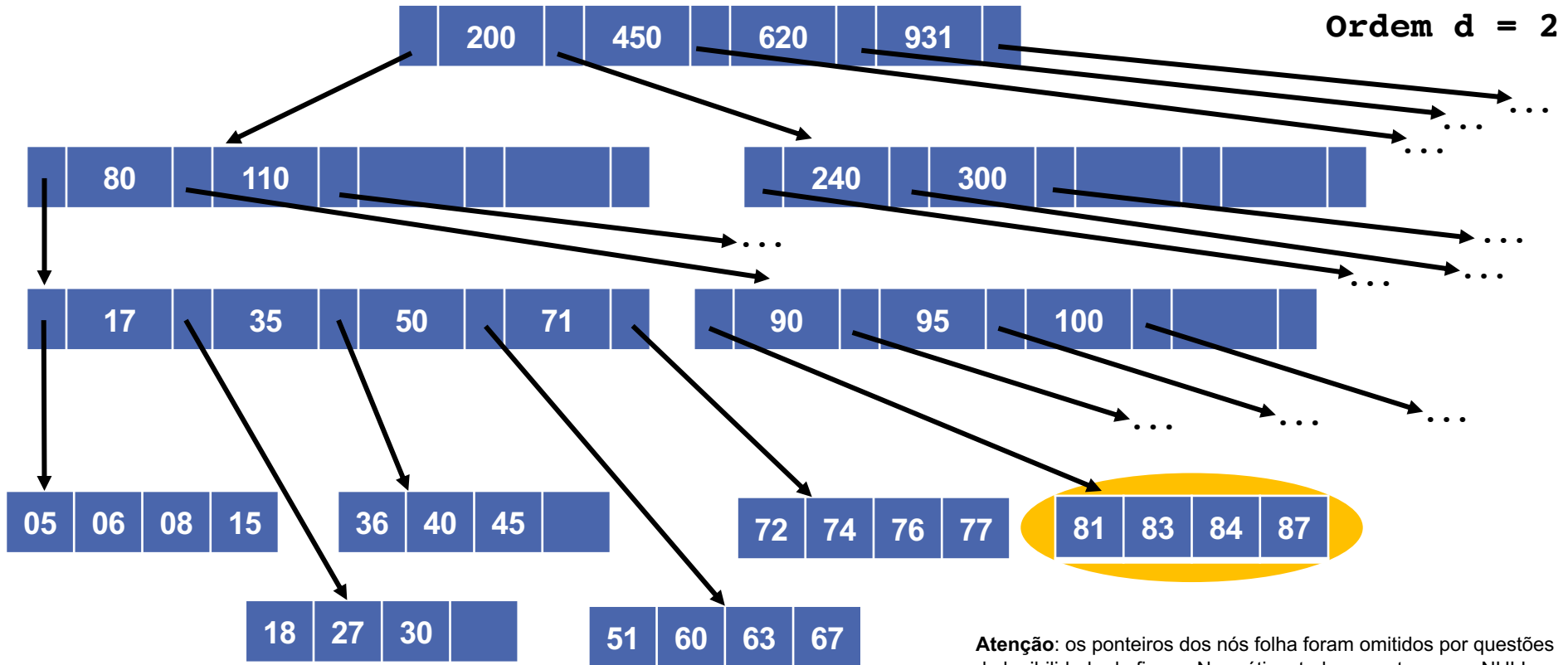
- Inserir a chave 85

Inserção

com Particionamento

Inserir chave 85

Inserção faria a página ficar com $2d + 1$ chaves
81; 83; 84; 85; 87

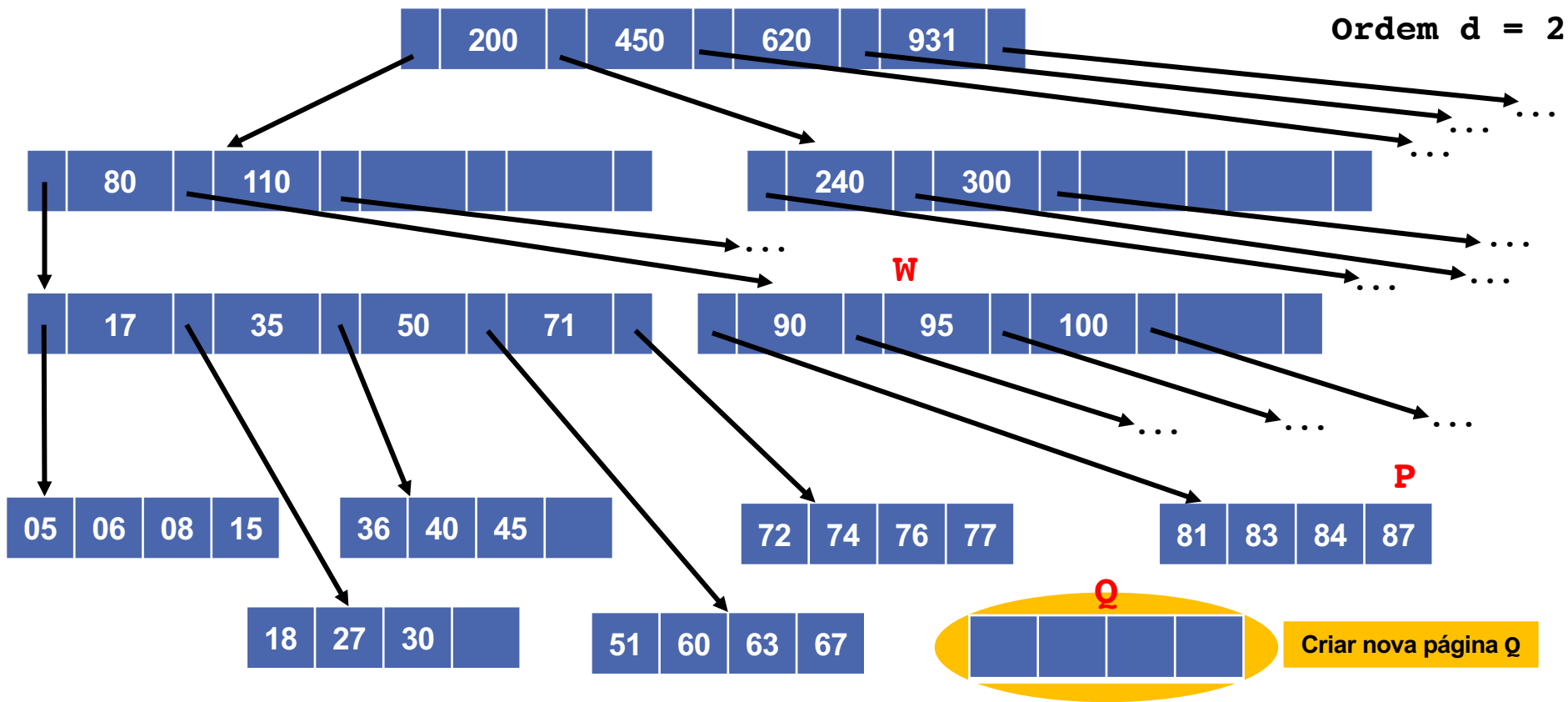


Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

Figura: Vanessa Braganholo (slides de árvore B)

Inserção

com Particionamento

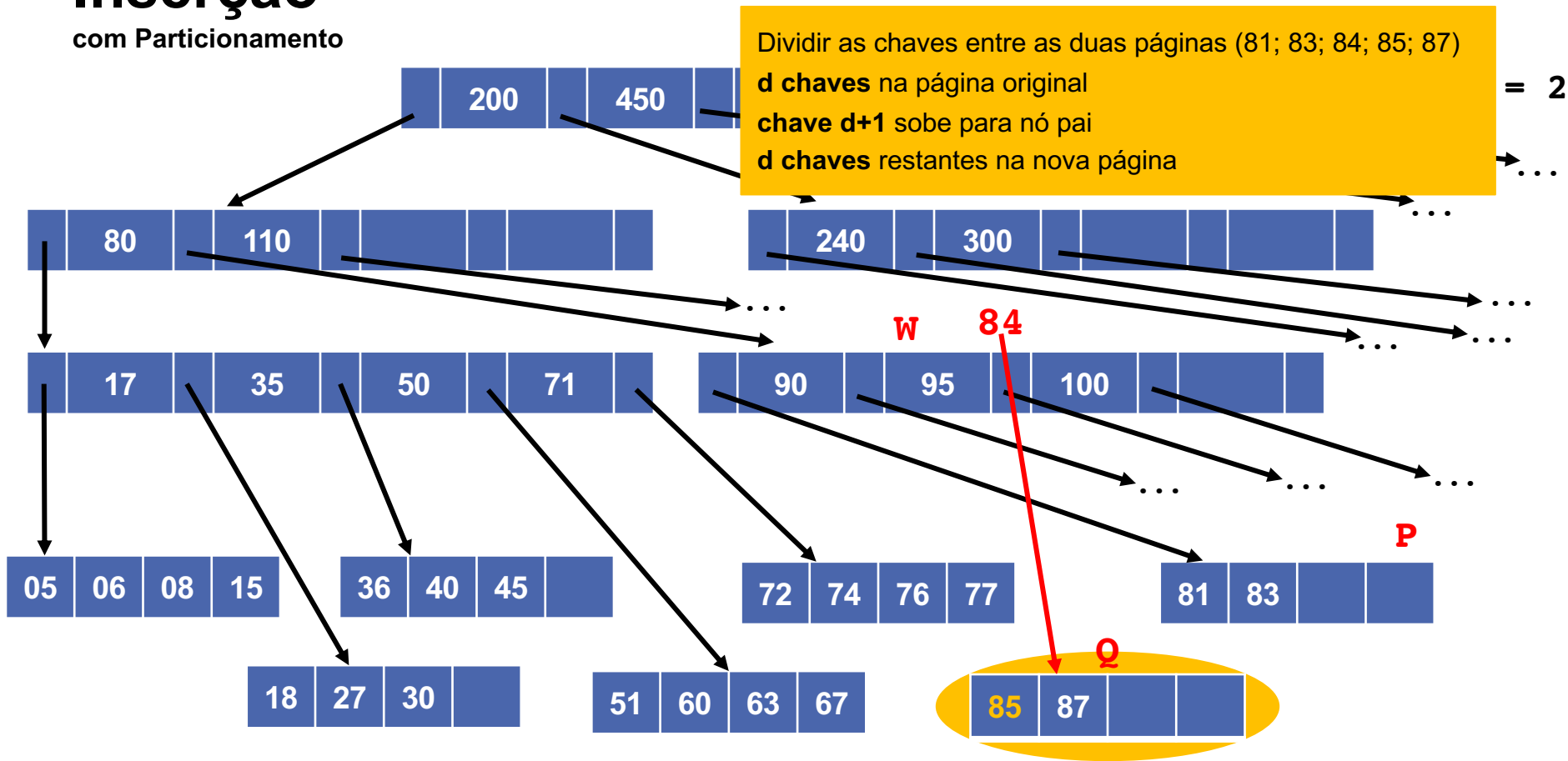


Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

Figura: Vanessa Braganholo (slides de árvore B)

Inserção

com Particionamento

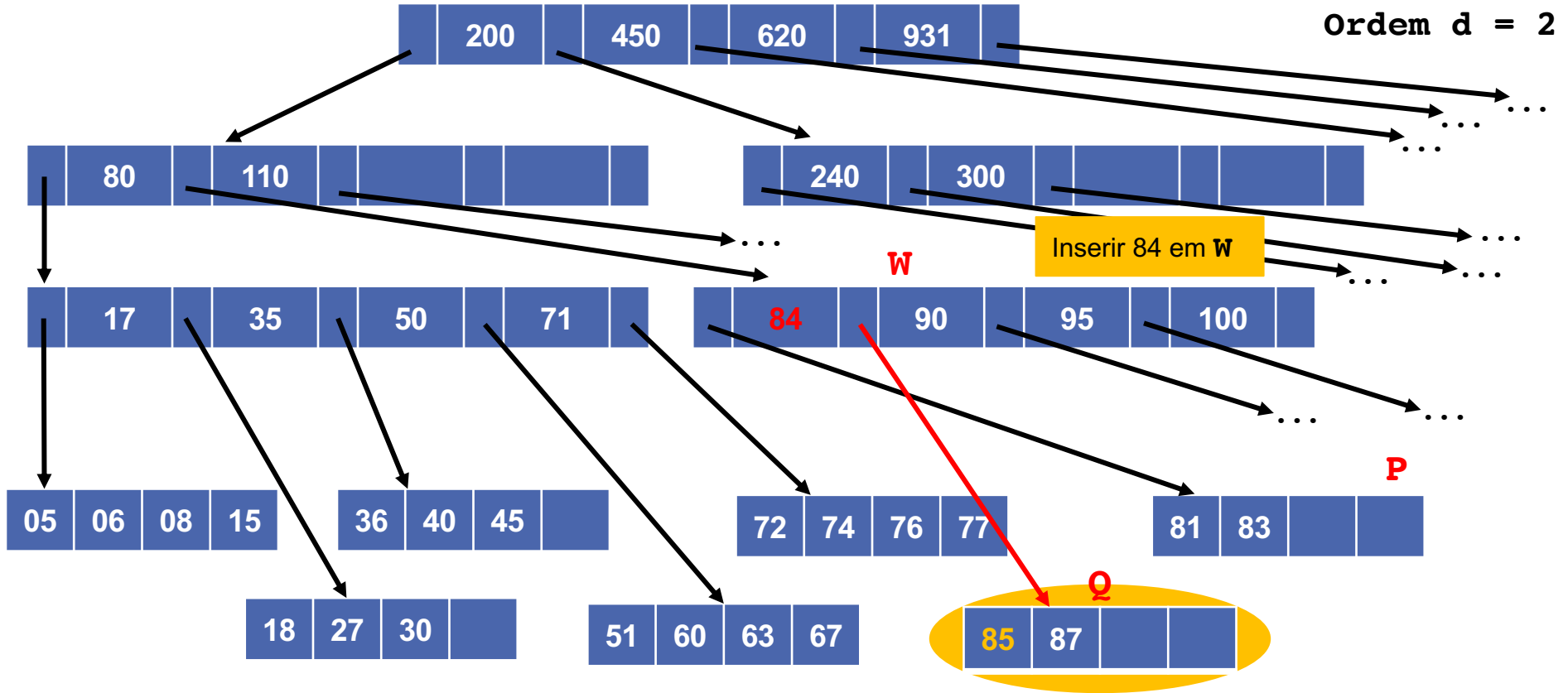


Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

Figura: Vanessa Braganholo (slides de árvore B)

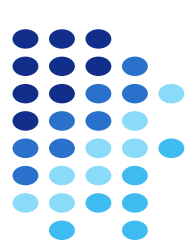
Inserção

com Particionamento



Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

Figura: Vanessa Braganholo (slides de árvore B)



Exemplo de Propagação

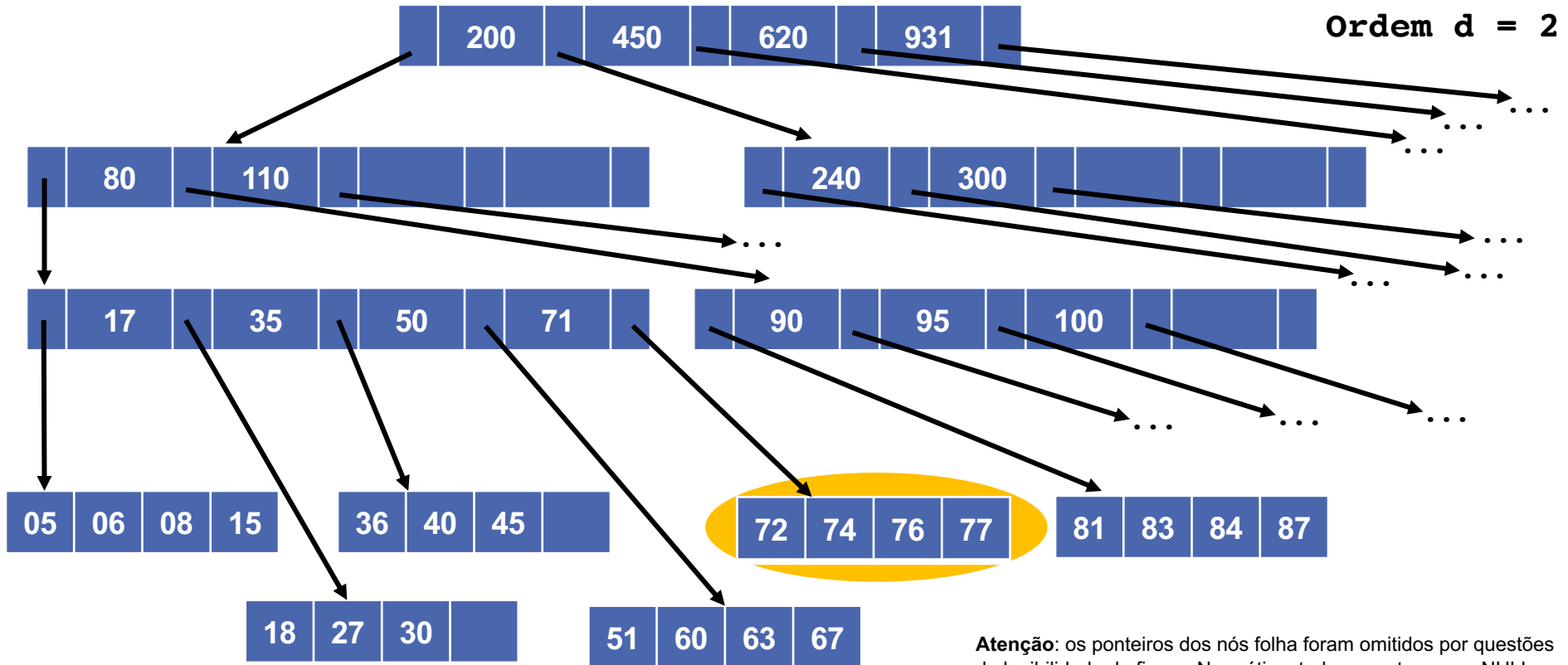
- Inserir a chave 73

Inserção

com Propagação

Inserir chave 73

Inserção faria a página ficar com $2d + 1$ chaves
72; 73; 74; 76; 77

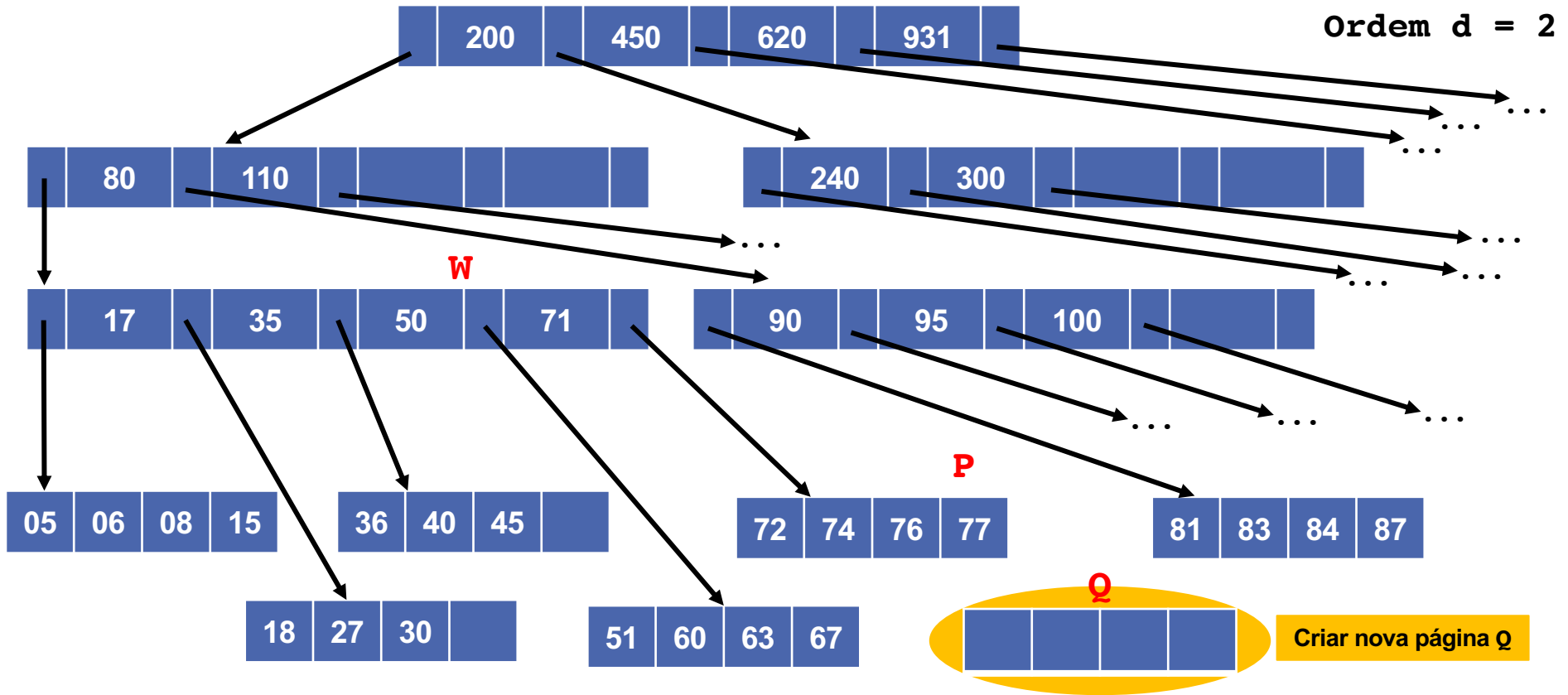


Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

Figura: Vanessa Braganholo (slides de árvore B)

Inserção

com Propagação

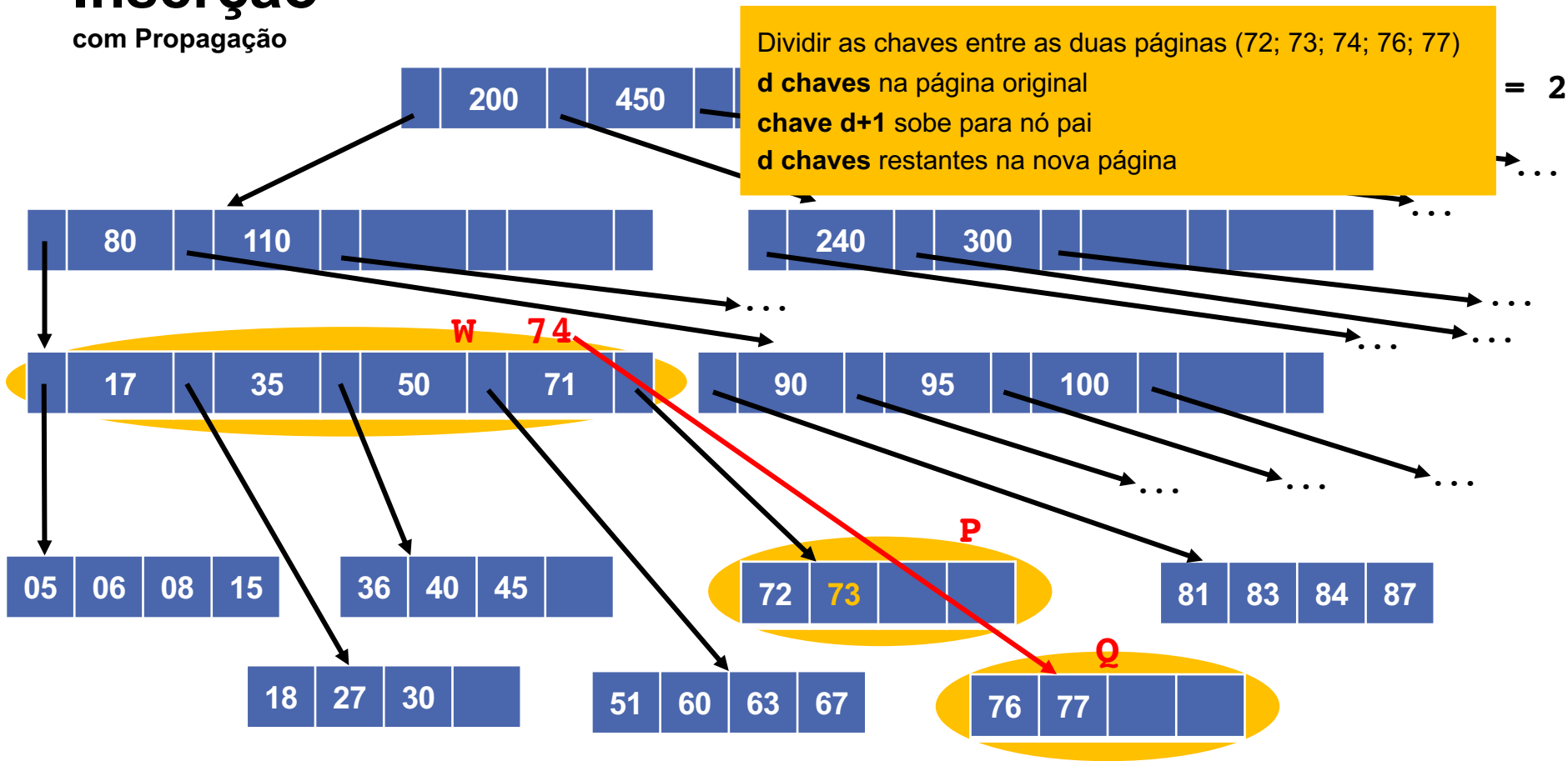


Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

Figura: Vanessa Braganholo (slides de árvore B)

Inserção

com Propagação

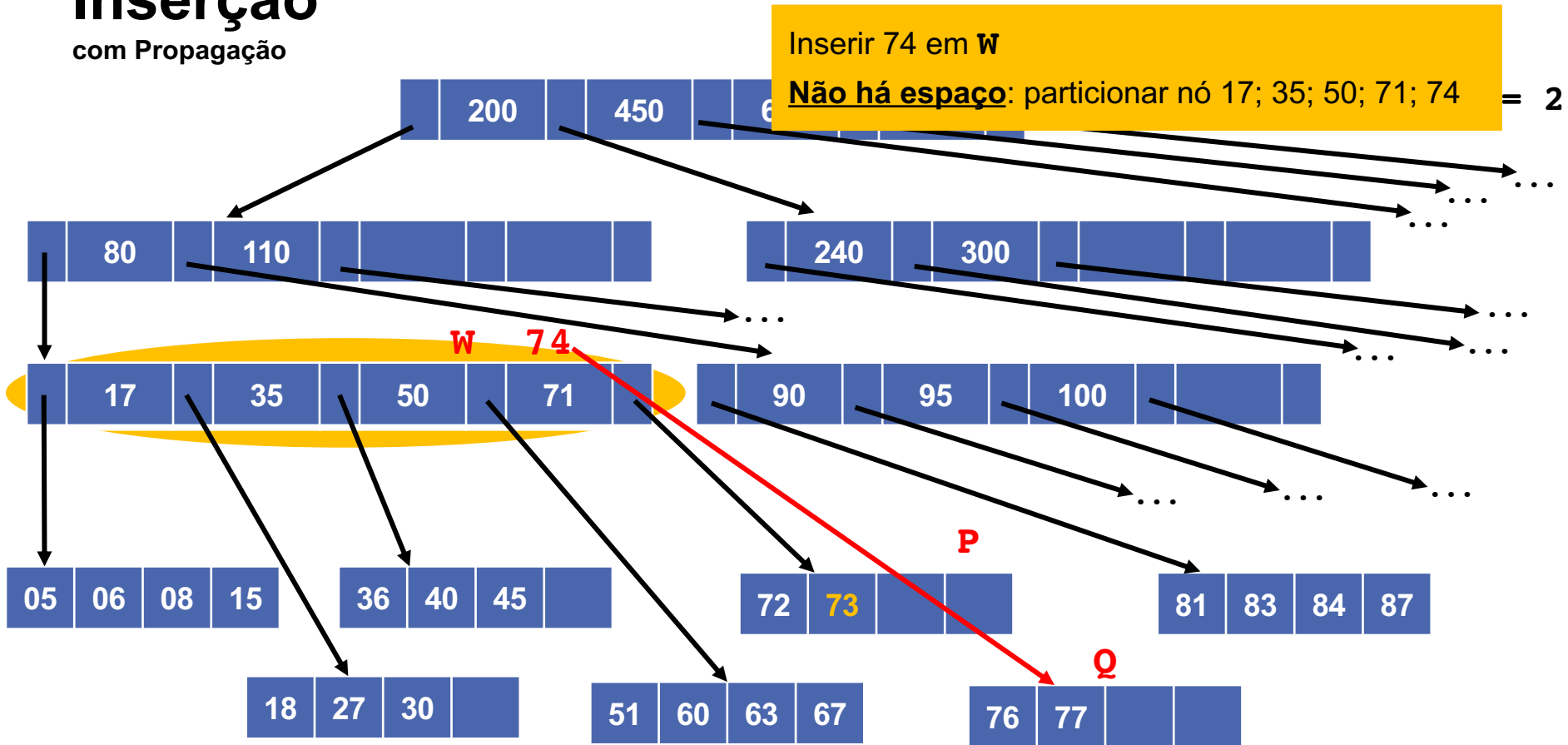


Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

Figura: Vanessa Braganholo (slides de árvore B)

Inserção

com Propagação

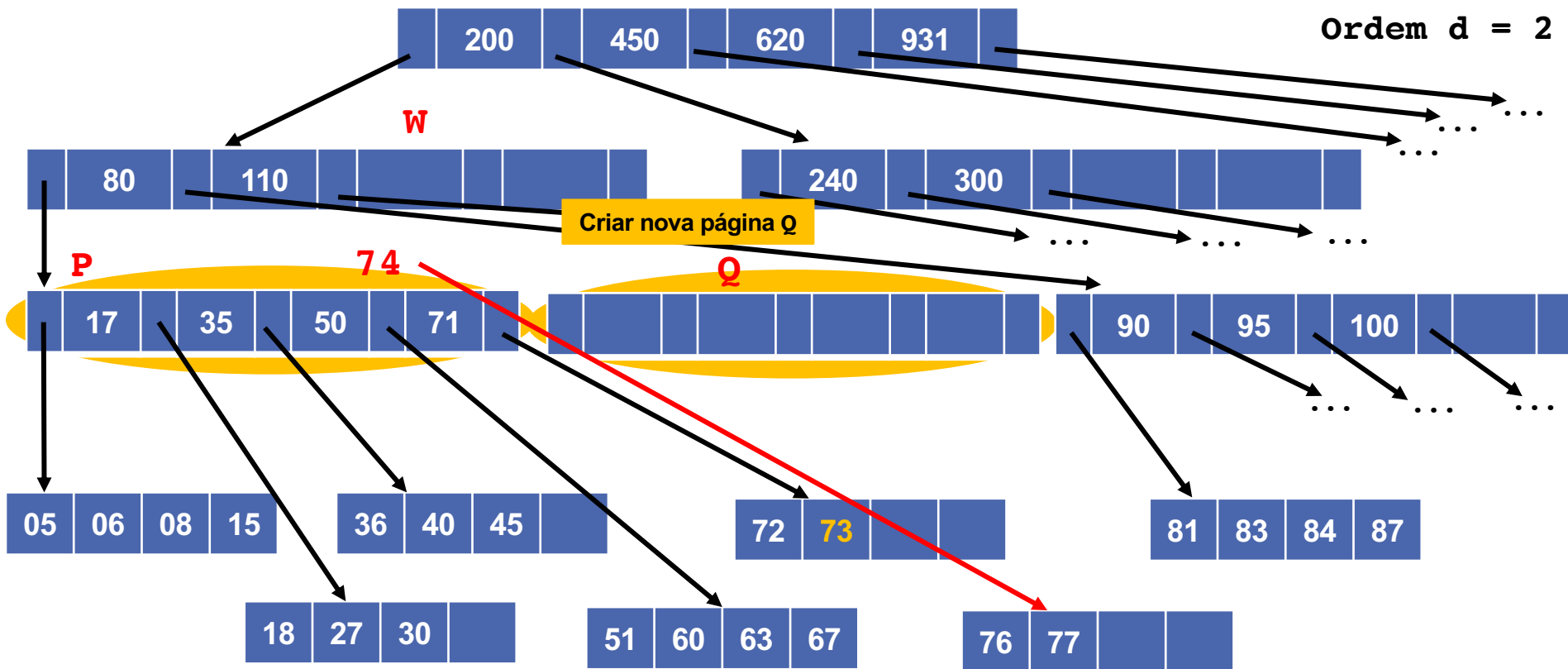


Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

Figura: Vanessa Braganholo (slides de árvore B)

Inserção

com Propagação

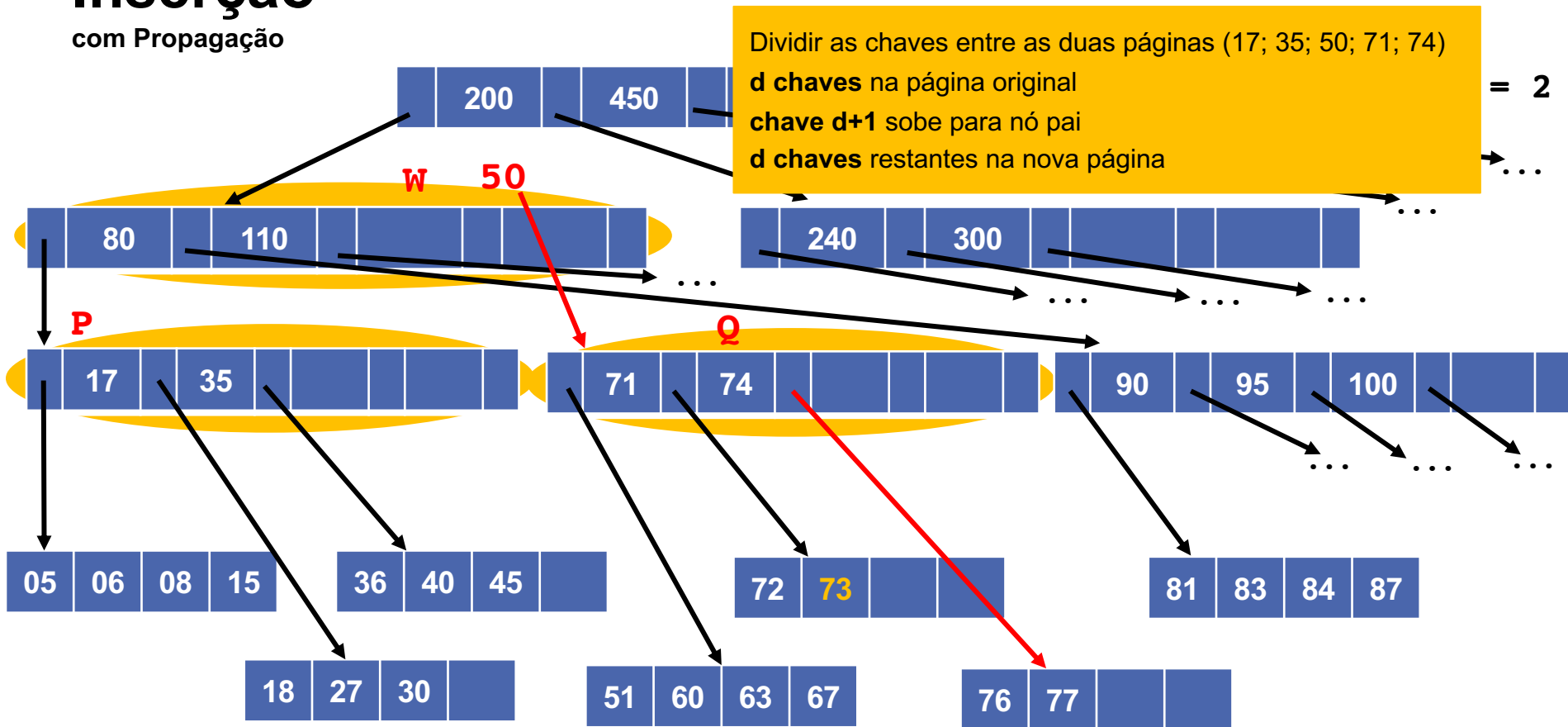


Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

Figura: Vanessa Braganholo (slides de árvore B)

Inserção

com Propagação

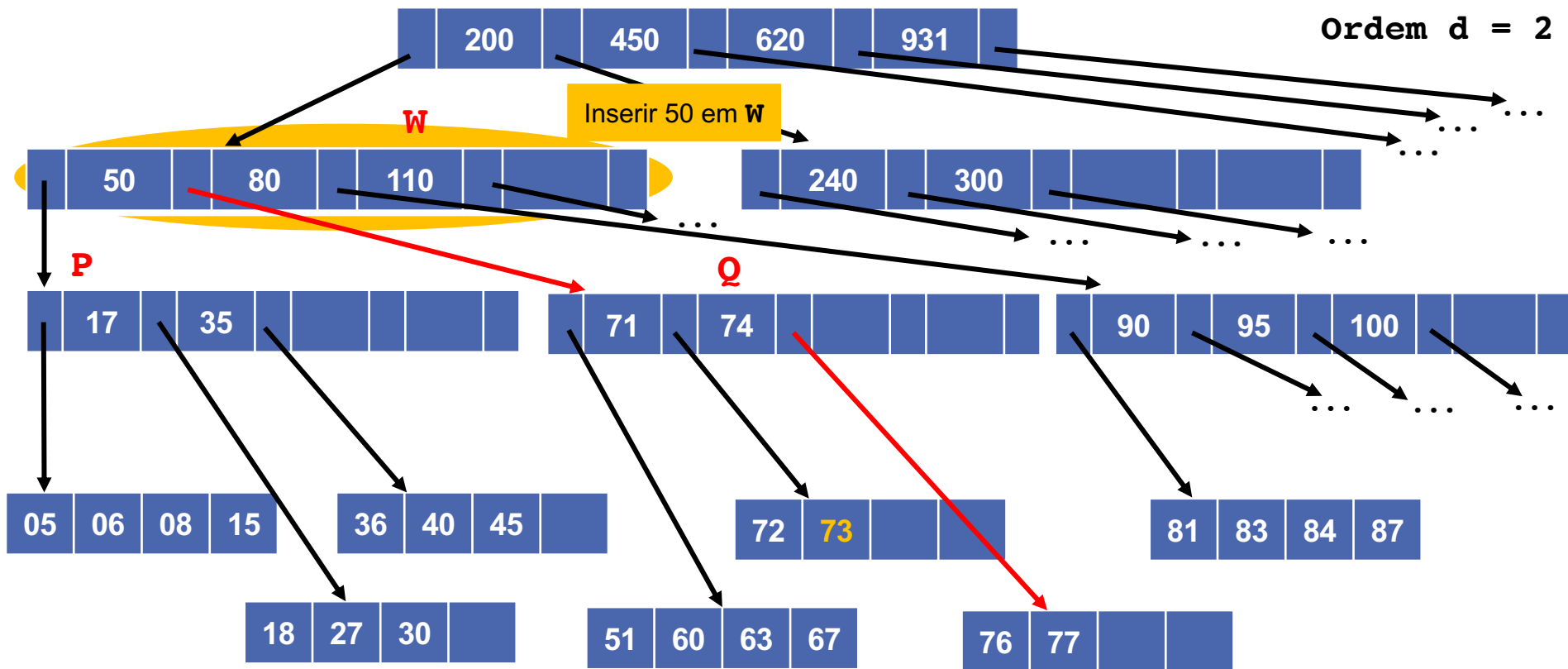


Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

Figura: Vanessa Braganholo (slides de árvore B)

Inserção

com Propagação

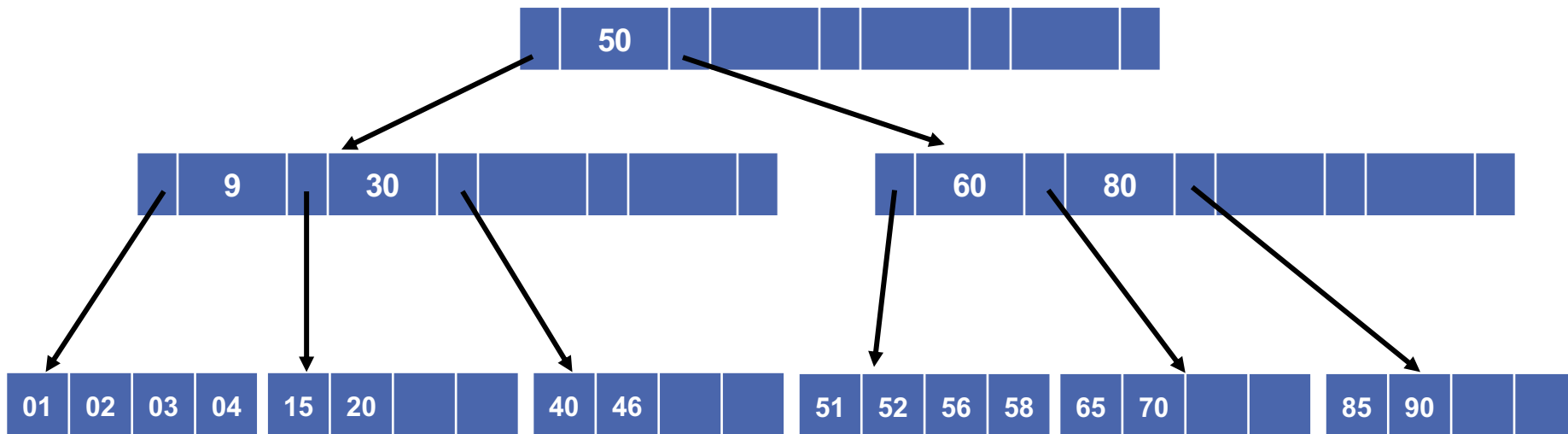


Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

Figura: Vanessa Braganholo (slides de árvore B)

Exercício:

Inserir as chaves 57, 71, 72, 73 (nessa ordem) na Árvore B abaixo:

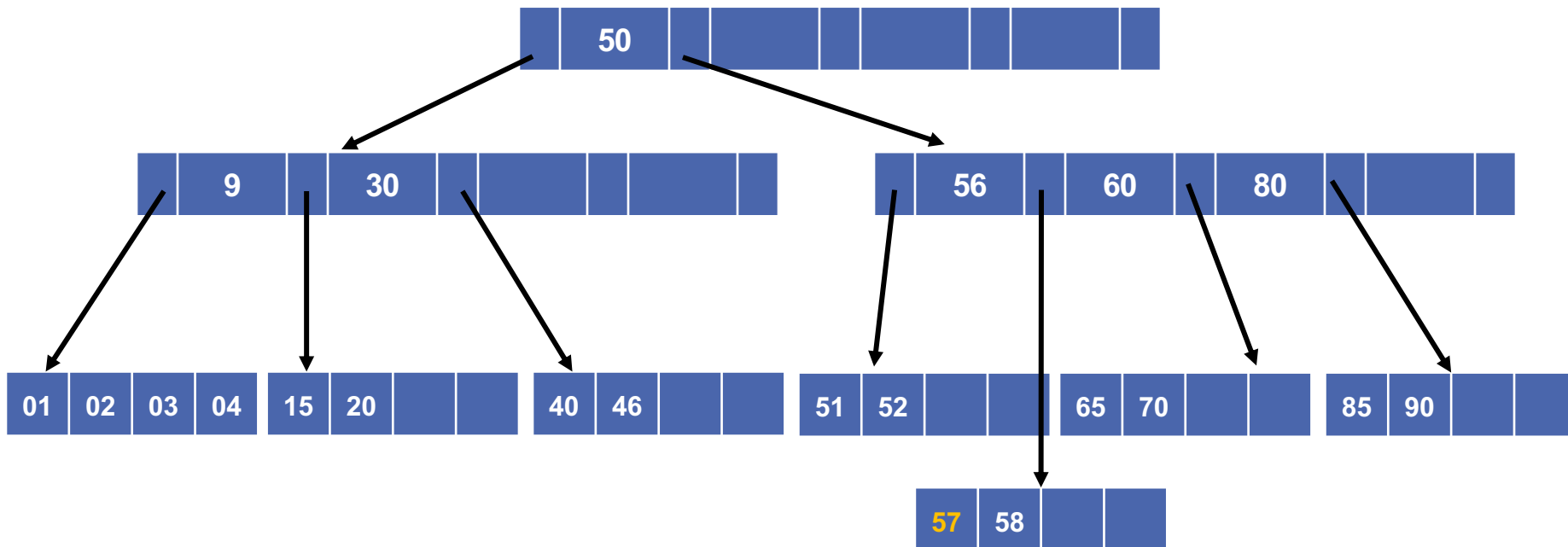


Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

Figura: Vanessa Braganholo (slides de árvore B)

Exercício:

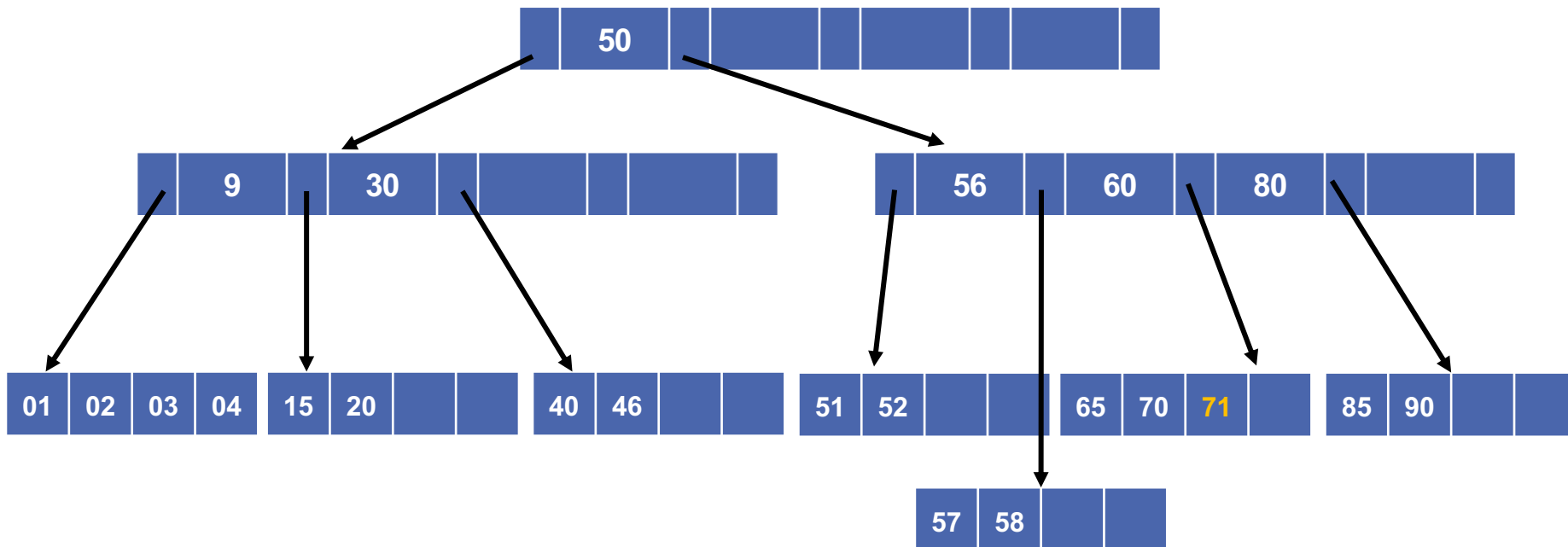
Inserir as chaves **57**, 71, 72, 73 (nessa ordem) na Árvore B abaixo:



Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

Exercício:

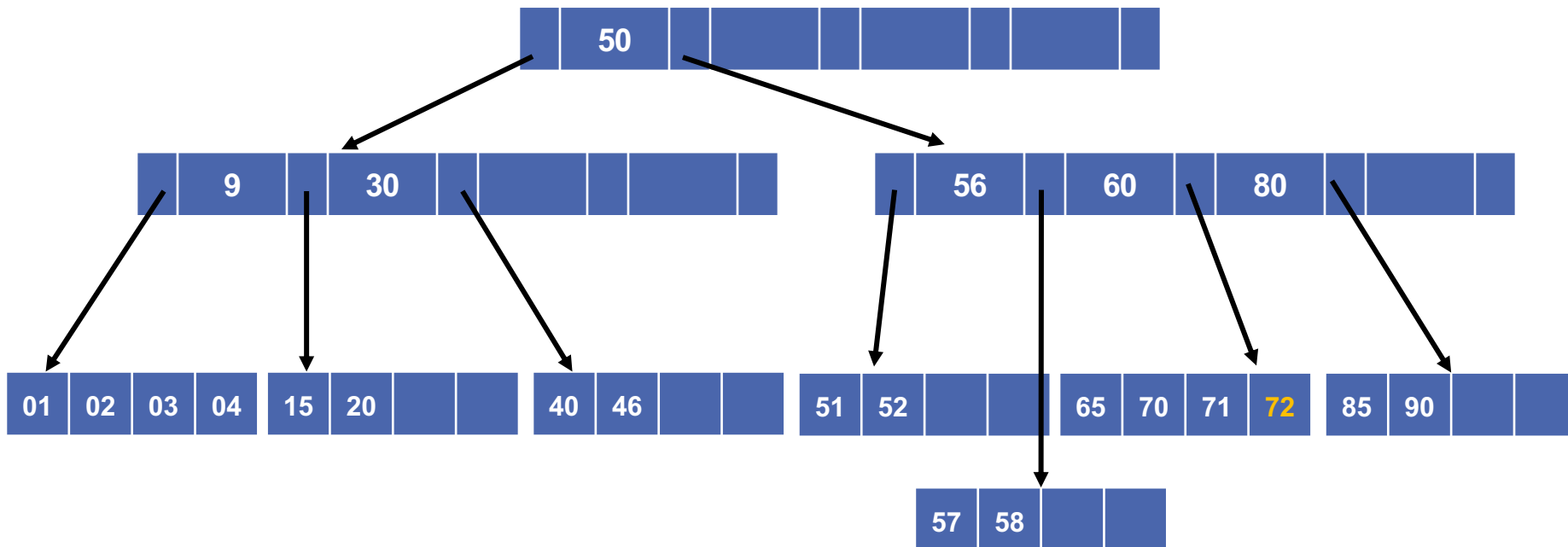
Inserir as chaves 57, 71, 72, 73 (nessa ordem) na Árvore B abaixo:



Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

Exercício:

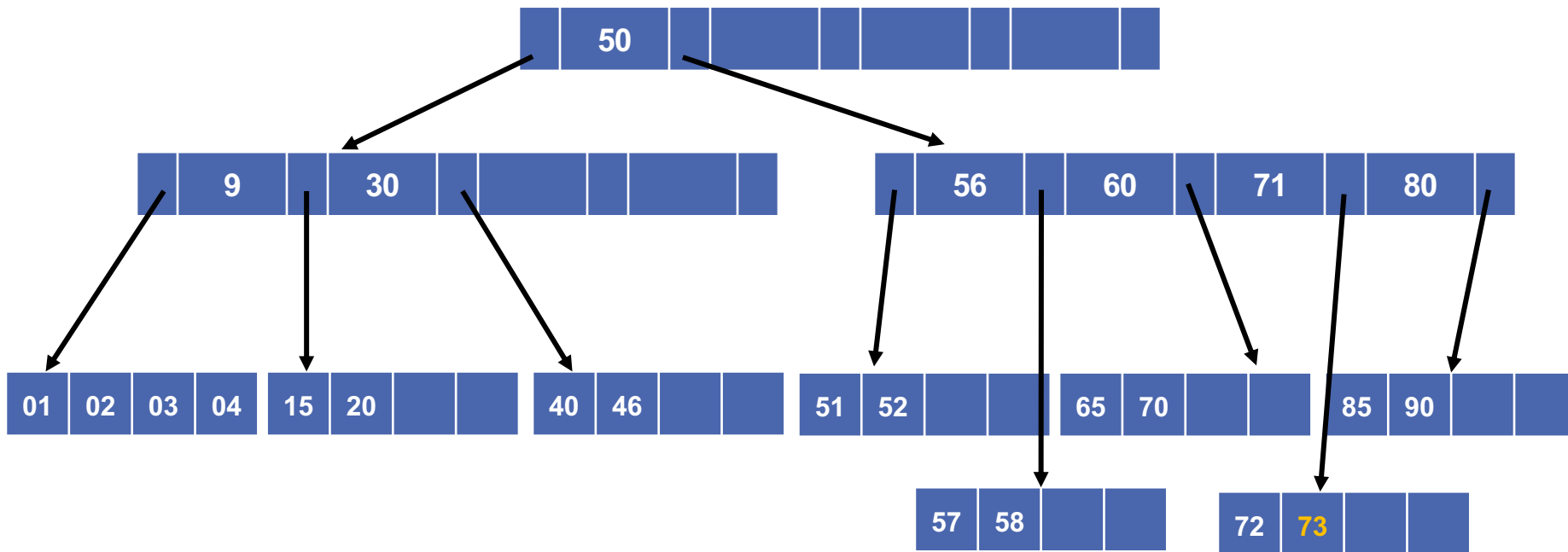
Inserir as chaves 57, 71, **72**, 73 (nessa ordem) na Árvore B abaixo:



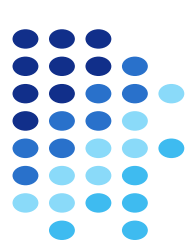
Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

Exercício:

Inserir as chaves 57, 71, 72, **73** (nessa ordem) na Árvore B abaixo:



Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

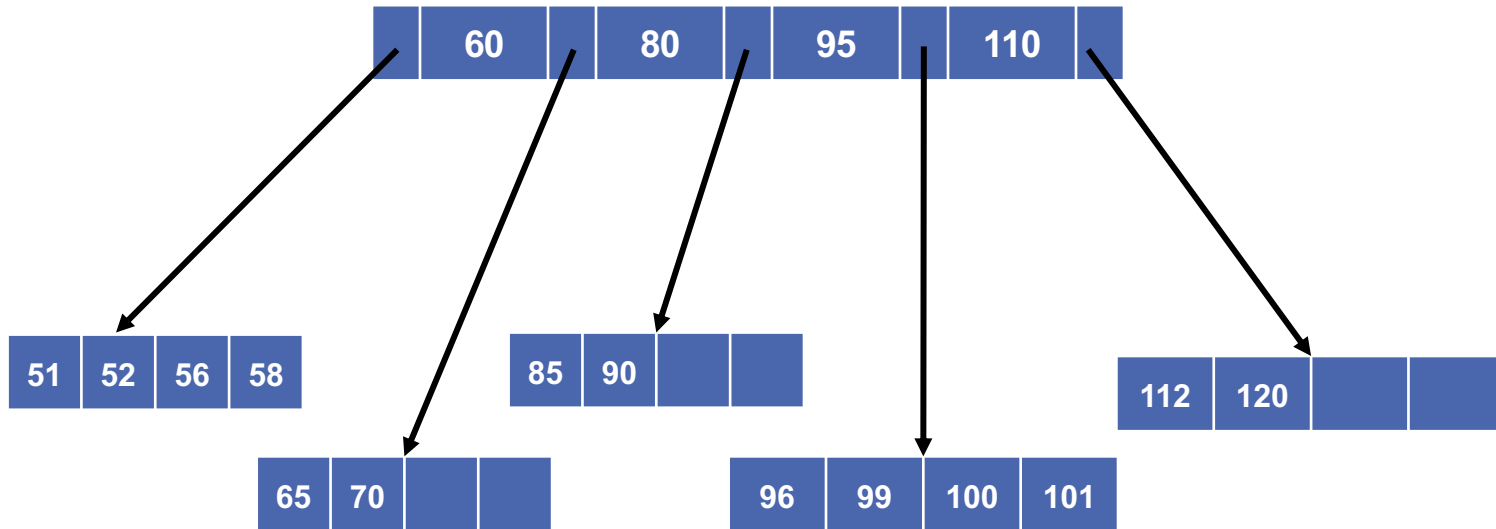


Divisão do Nó Raiz

- Em alguns casos o particionamento se propaga para a raiz
- Nesse caso, o nó raiz é particionado normalmente, mas, como a raiz não tem pai, cria-se um novo nó, que passa a ser a nova raiz

Exemplo:

Inserir a chave 97

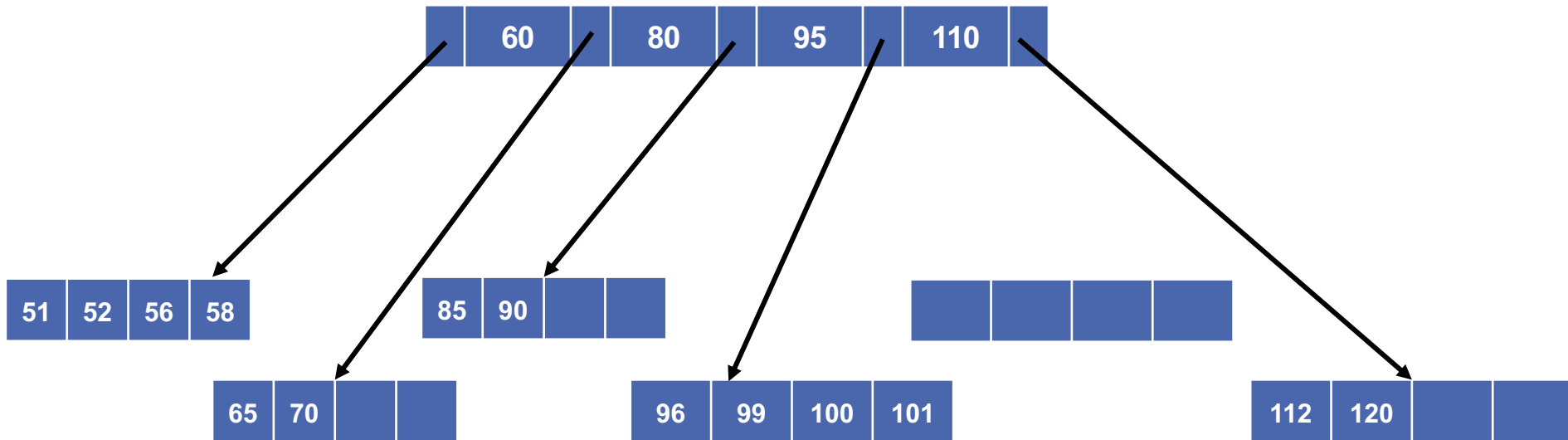


Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

Figura: Vanessa Braganholo (slides de árvore B)

Exemplo:

Inserir a chave 97

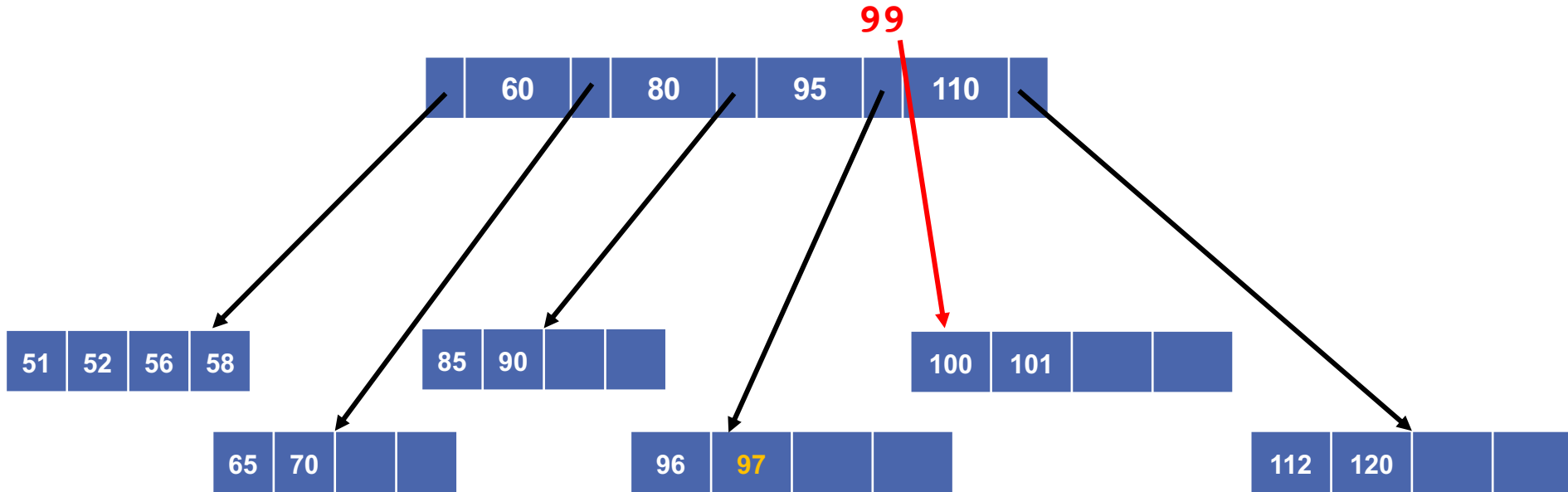


Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

Figura: Vanessa Braganholo (slides de árvore B)

Exemplo:

Inserir a chave 97

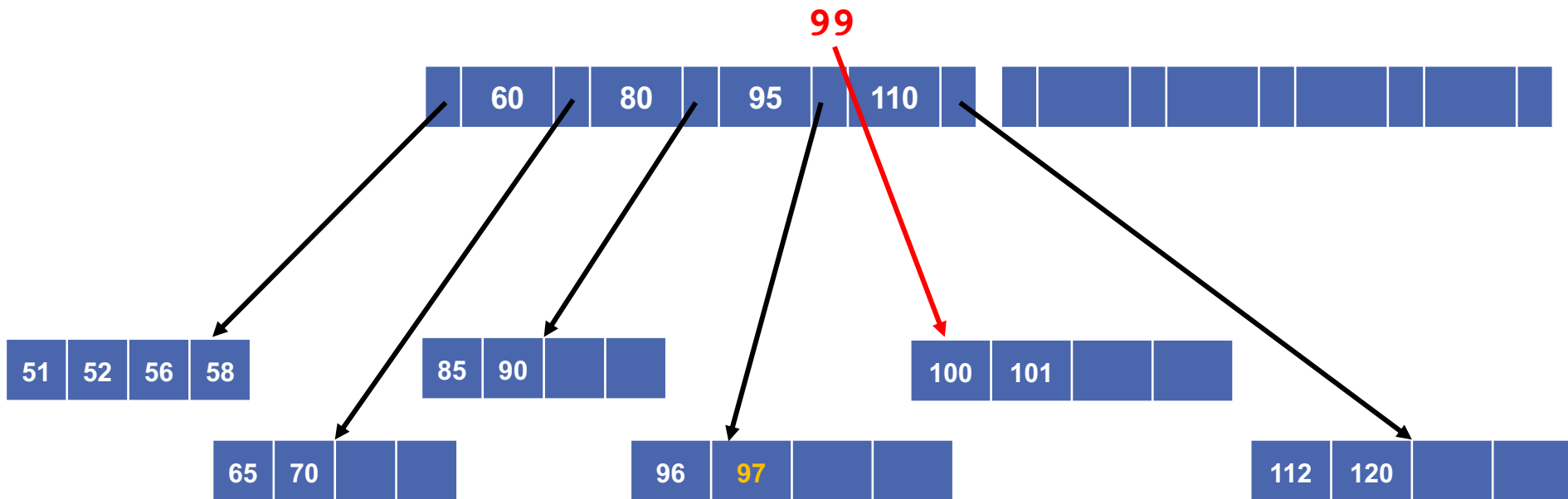


Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

Figura: Vanessa Braganholo (slides de árvore B)

Exemplo:

Inserir a chave 97

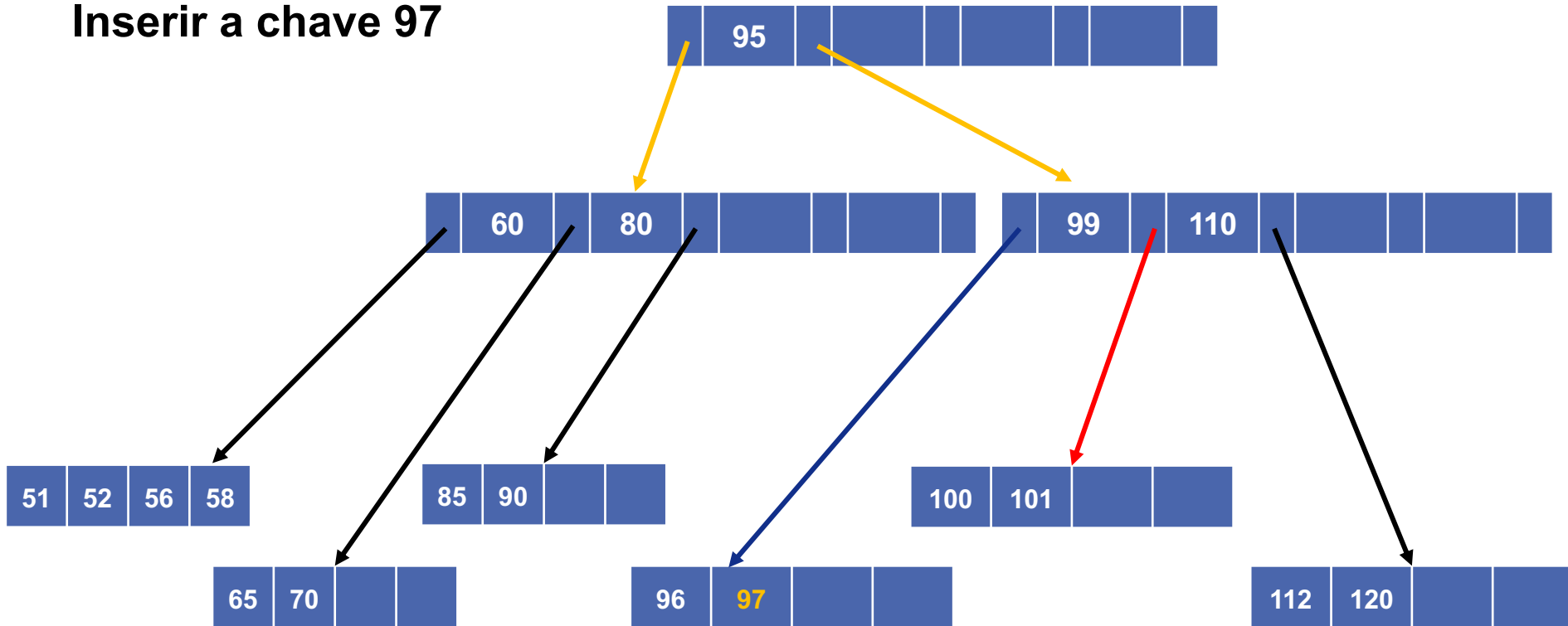


Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

Figura: Vanessa Braganholo (slides de árvore B)

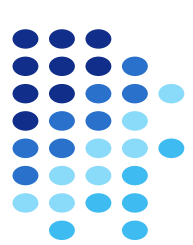
Exemplo:

Inserir a chave 97



Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

Figura: Vanessa Braganholo (slides de árvore B)

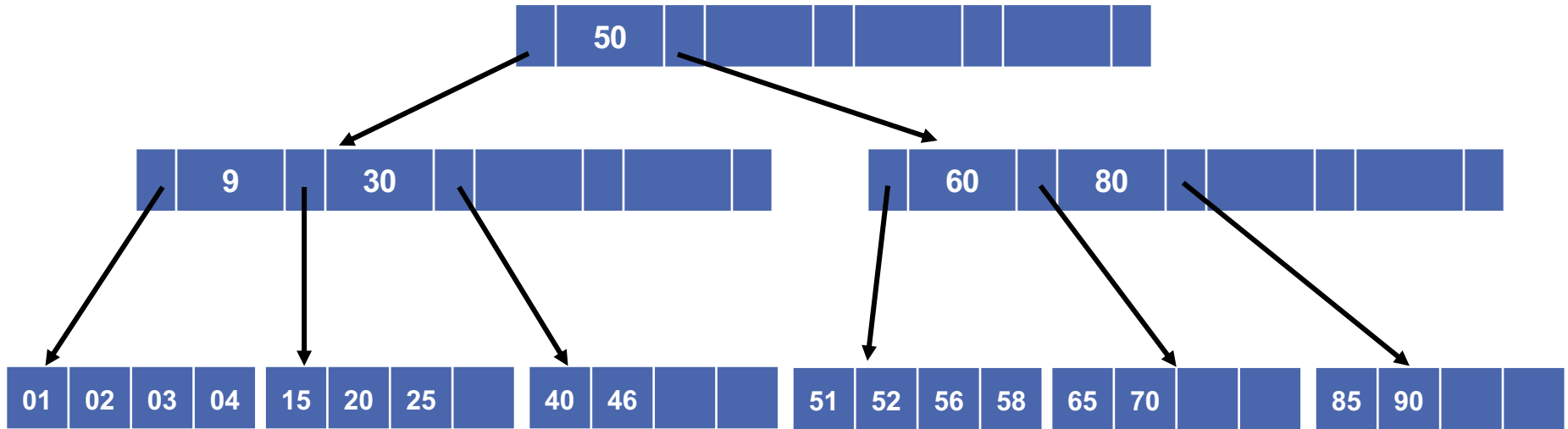


Exclusão

- Duas situações possíveis:
 - A entrada **x** está em um nó folha
 - Neste caso, simplesmente remover a entrada **x**
- A entrada **x** não está em um nó folha
 - Substituir **x** pela chave **y** imediatamente maior
 - Note que **y** necessariamente pertence a uma folha, pela forma como a **árvore B** é estruturada

Exemplo:

Exclusão da chave 03 (nó folha)

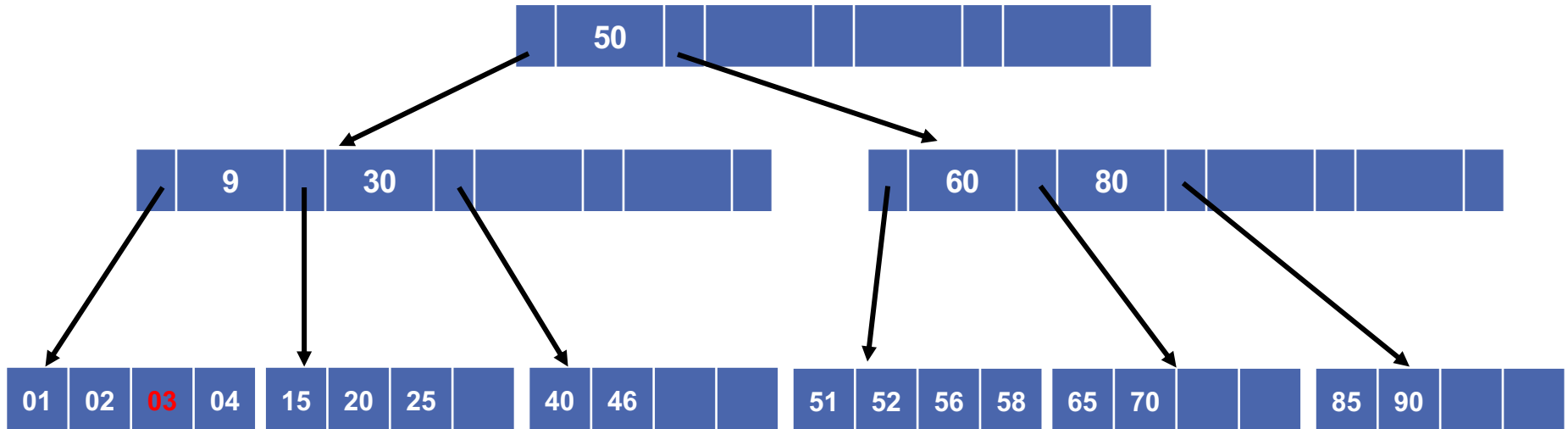


Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

Figura: Vanessa Braganholo (slides de árvore B)

Exemplo:

Exclusão da chave 03 (nó folha)

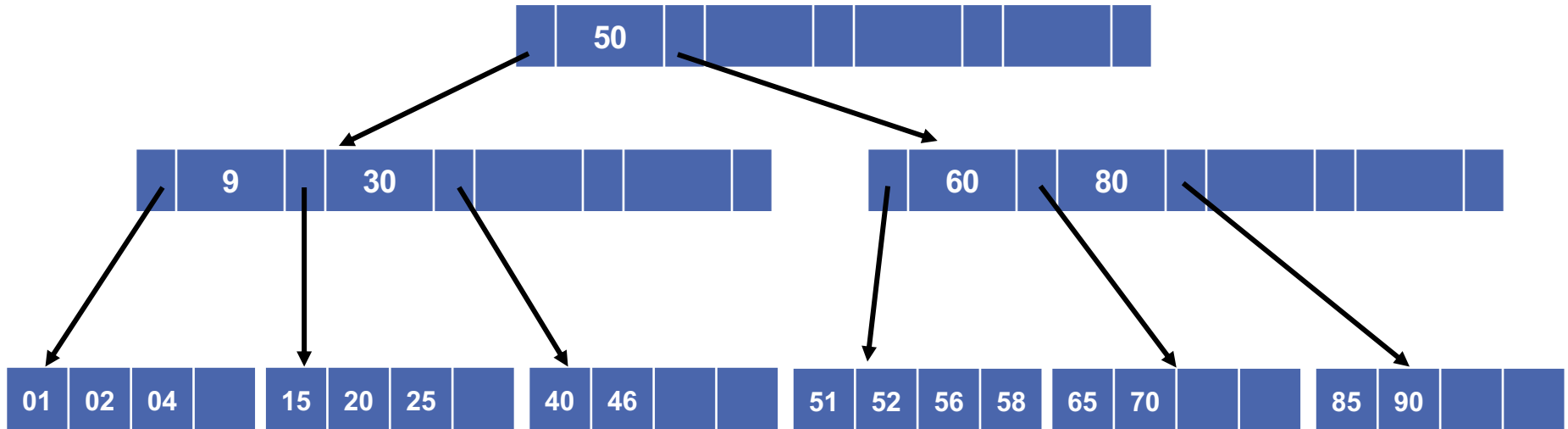


Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

Figura: Vanessa Braganholo (slides de árvore B)

Exemplo:

Exclusão da chave 03 (nó folha)

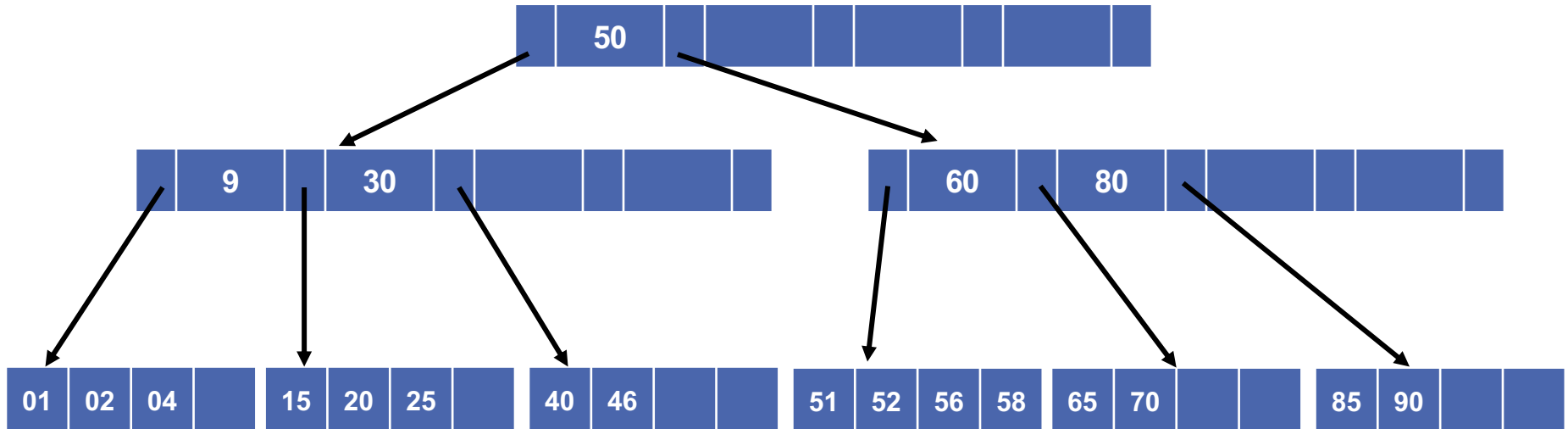


Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

Figura: Vanessa Braganholo (slides de árvore B)

Exemplo:

Exclusão da chave 09 (nó não folha)



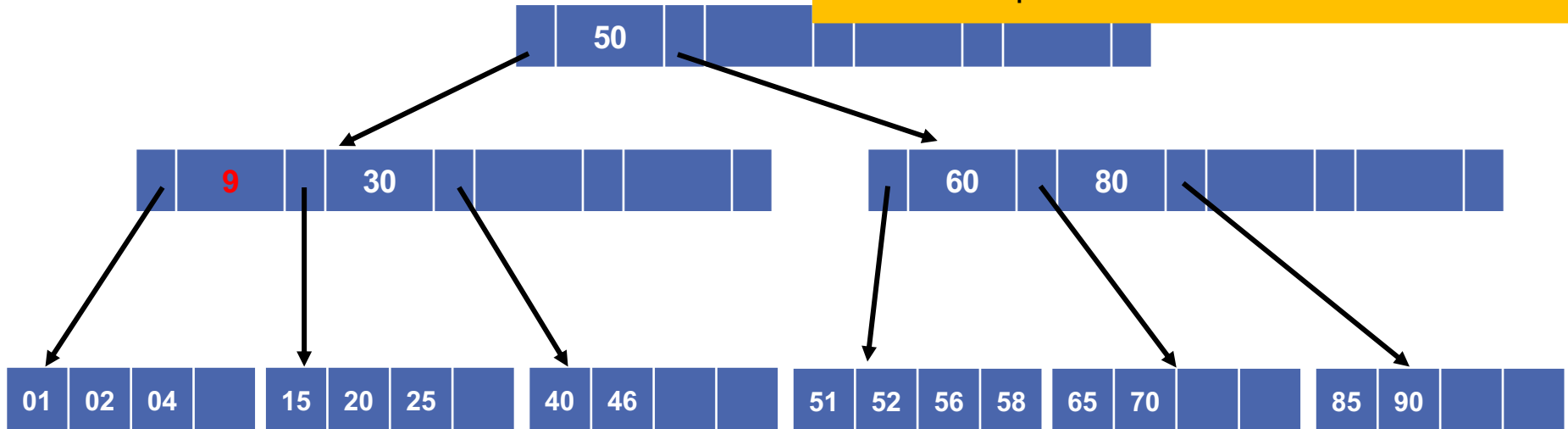
Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

Figura: Vanessa Braganholo (slides de árvore B)

Exemplo:

Exclusão da chave 09 (nó não folha)

Substituir pela chave imediatamente maior



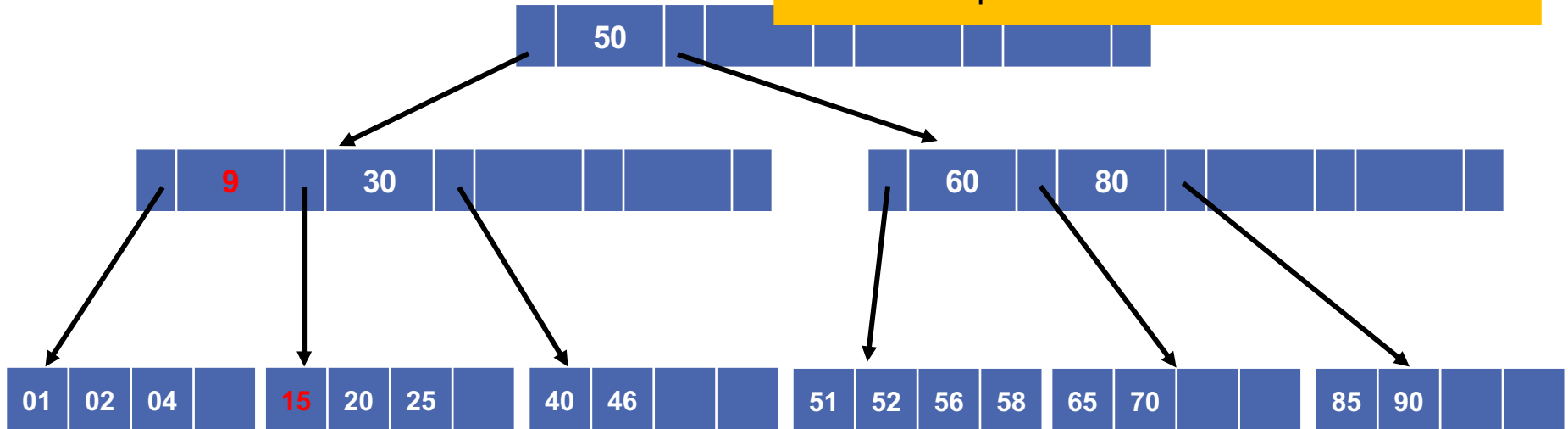
Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

Figura: Vanessa Braganholo (slides de árvore B)

Exemplo:

Exclusão da chave 09 (nó não folha)

Substituir pela chave imediatamente maior



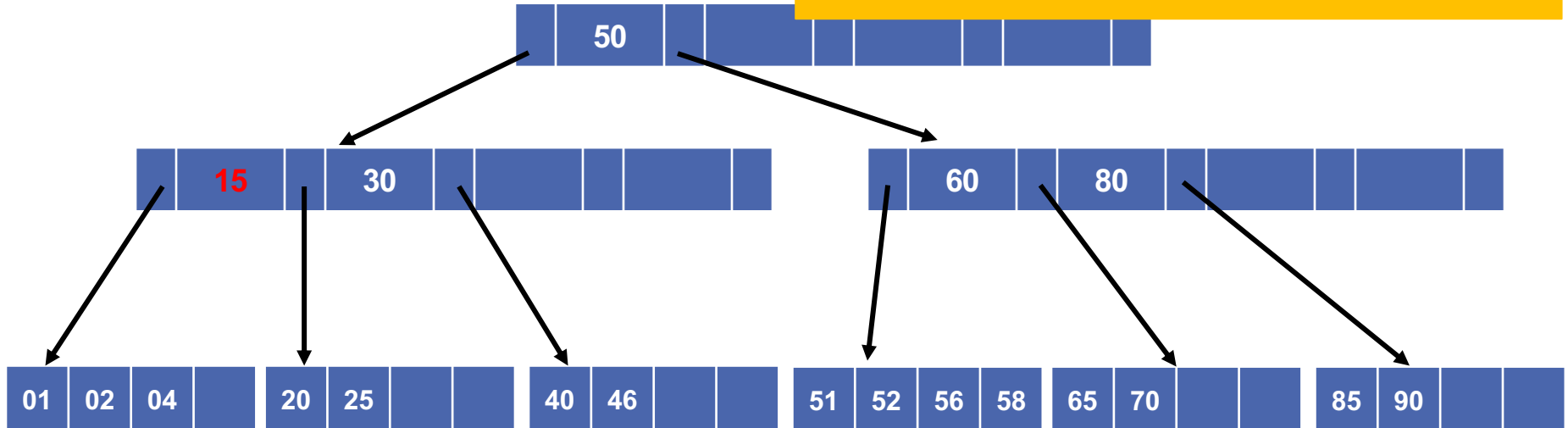
Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

Figura: Vanessa Braganholo (slides de árvore B)

Exemplo:

Exclusão da chave 09 (nó não folha)

Substituir pela chave imediatamente maior



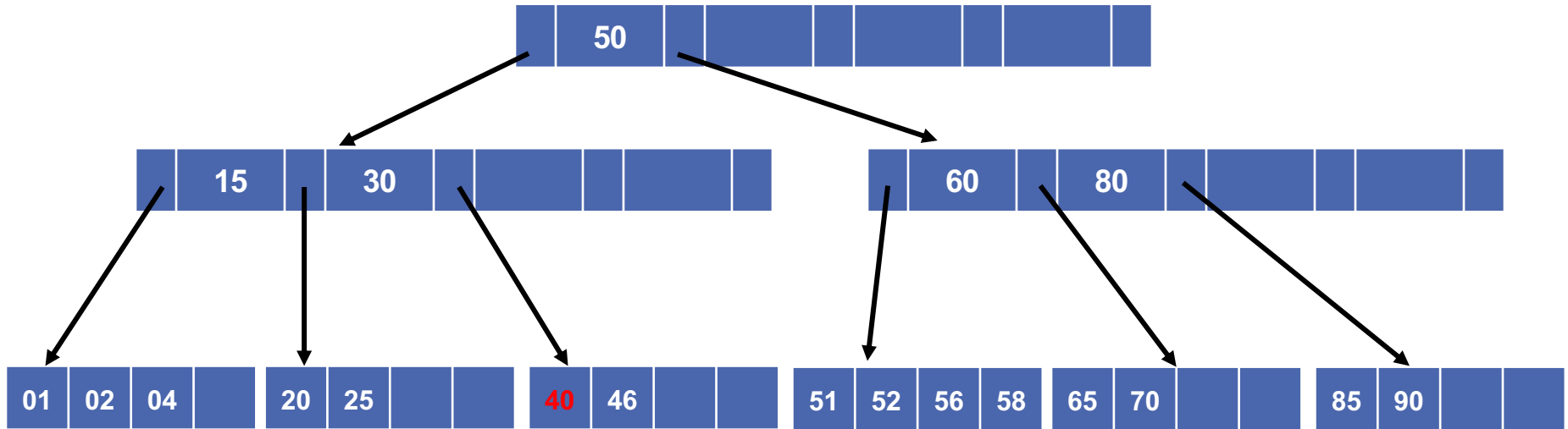
Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

Figura: Vanessa Braganholo (slides de árvore B)

Exemplo:

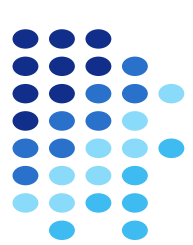
Exclusão da chave 40

Problema: o nó ficaria com menos de **d** chaves, o que não é permitido



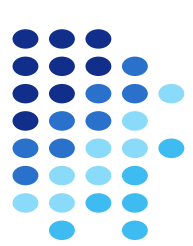
Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

Figura: Vanessa Braganholo (slides de árvore B)



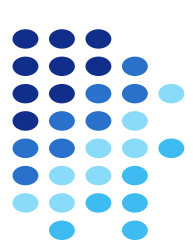
Solução:

- Concatenação ou Redistribuição



Concatenação

- Duas páginas **P** e **Q** são irmãs adjacentes se têm o mesmo pai **W** e são apontadas por dois ponteiros adjacentes em **W**
- **P** e **Q** podem ser **concatenadas** se:
 - são **irmãs adjacentes**; e
 - juntas possuem menos de **2d** chaves

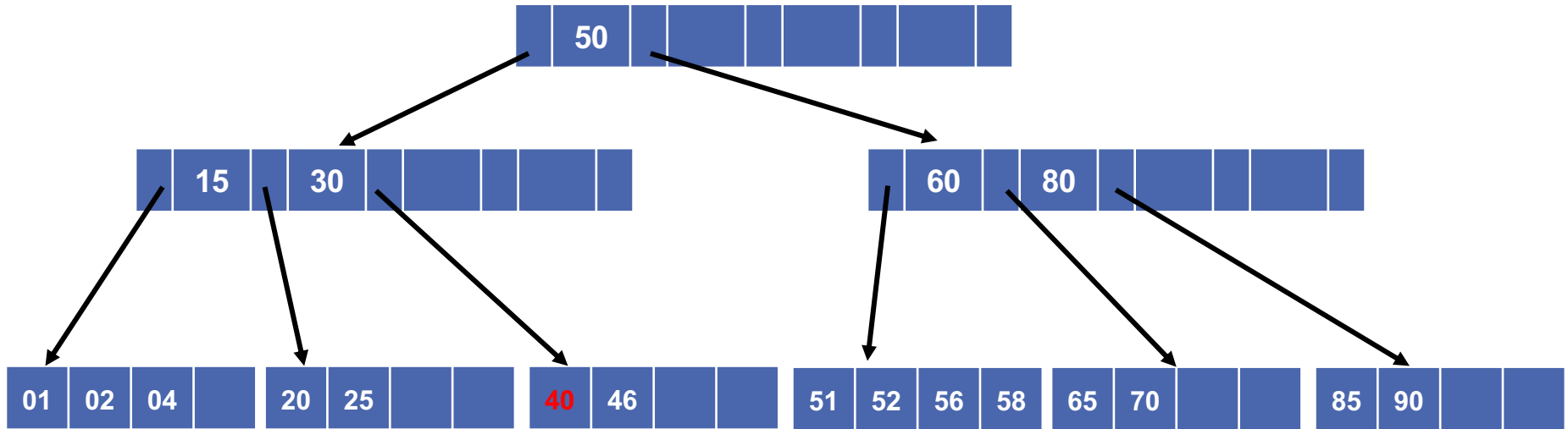


Operação de Concatenação de **P** e **Q**

- Agrupar as entradas de **Q** em **P**
- Em **W**, pegar a chave s_i que está entre os ponteiros que apontam para **P** e **Q**, e transferi-la para **P**
- Em **W**, eliminar o ponteiro p_i (ponteiro que ficava junto à chave s_i que foi transferida)

Exemplo:

Exclusão da chave 40



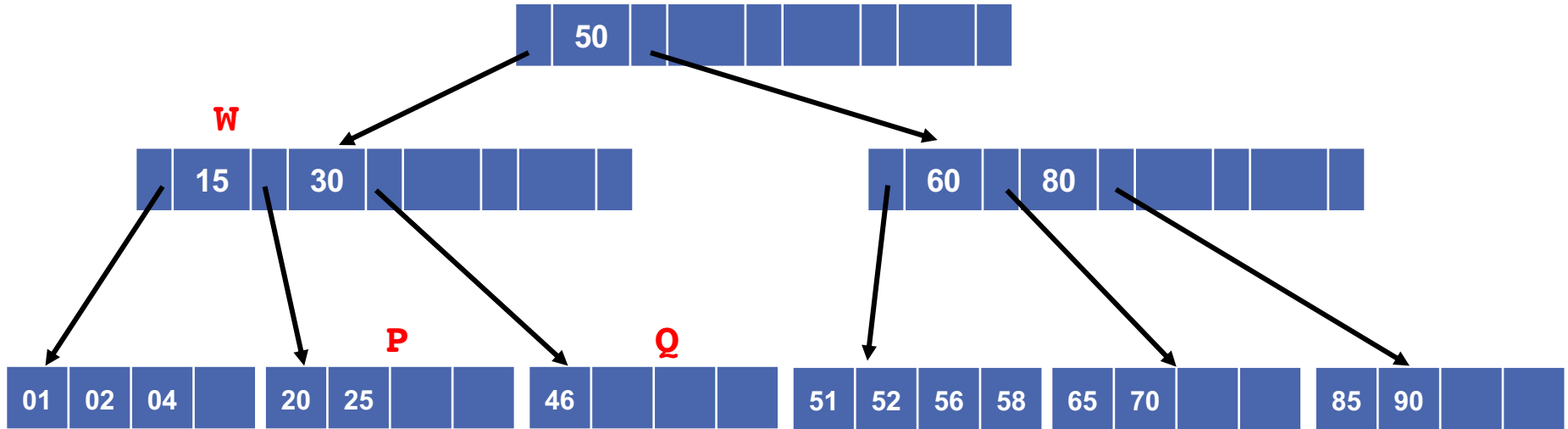
Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

Figura: Vanessa Braganholo (slides de árvore B)

Exemplo:

Exclusão da chave 40

Página **Q** ficou com menos de **d** chaves
Página **P** e **Q** são irmãs adjacentes
Soma de chaves de **P** e **Q** < $2d$
CONCATENAR P e Q

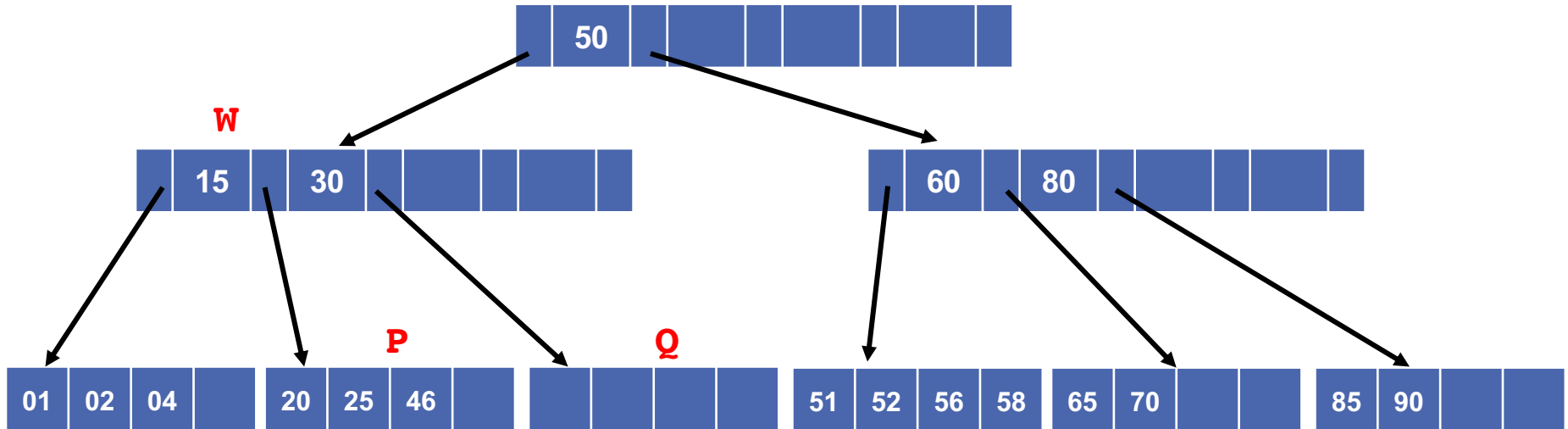


Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

Figura: Vanessa Braganholo (slides de árvore B)

Exemplo:

Exclusão da chave 40



Transferir dados de Q para P

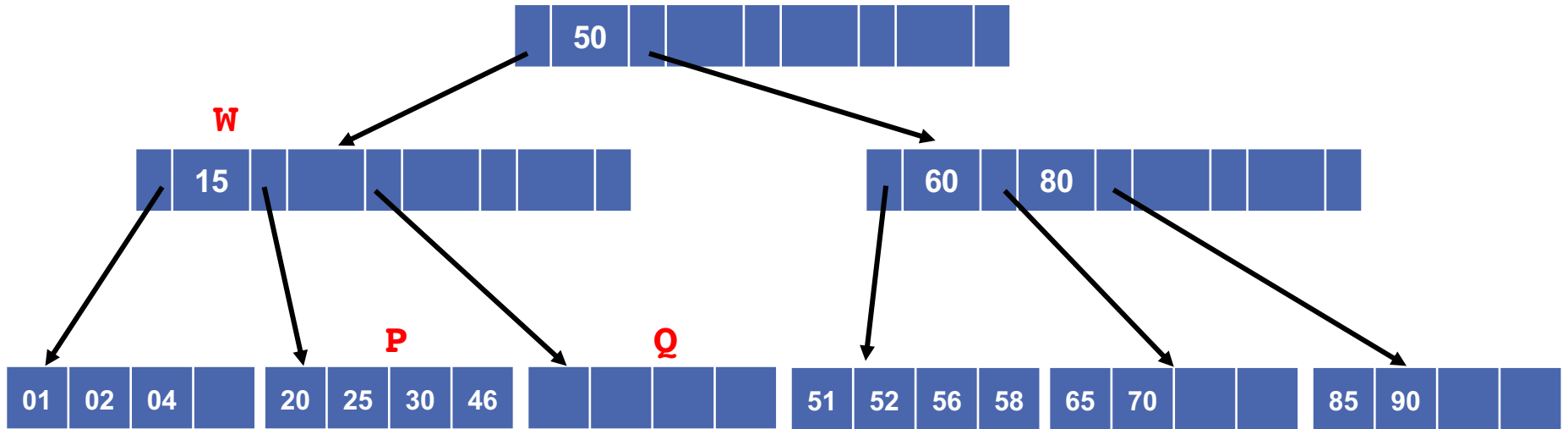
Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

Figura: Vanessa Braganholo (slides de árvore B)

Exemplo:

Exclusão da chave 40

Transferir chave que separa os ponteiros de **P** e **Q** em **W** para **P**



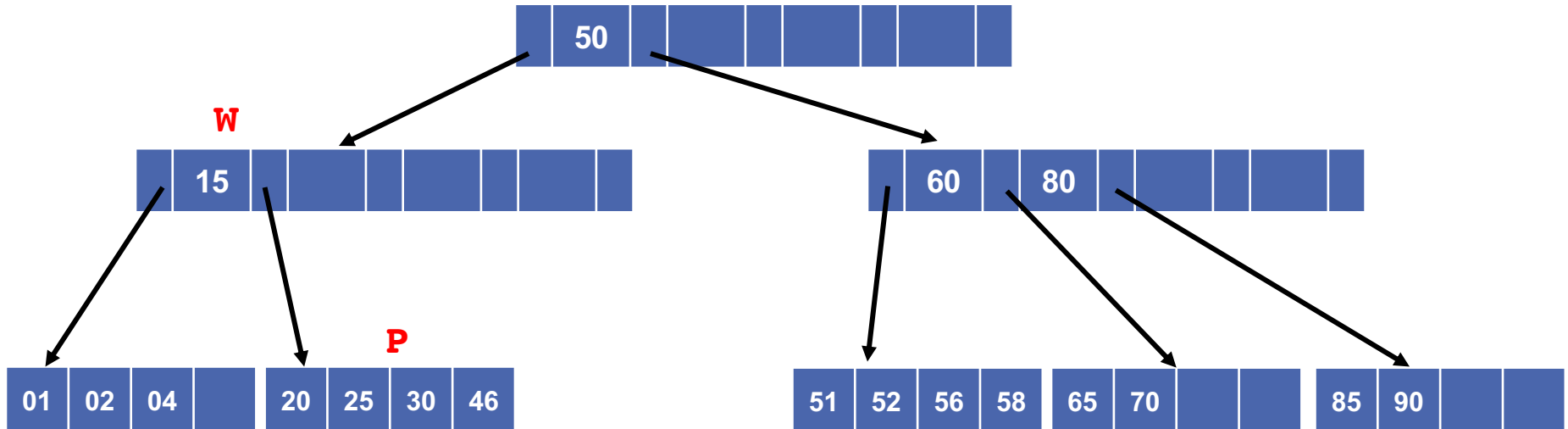
Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

Figura: Vanessa Braganholo (slides de árvore B)

Exemplo:

Exclusão da chave 40

Eliminar página Q e ponteiro



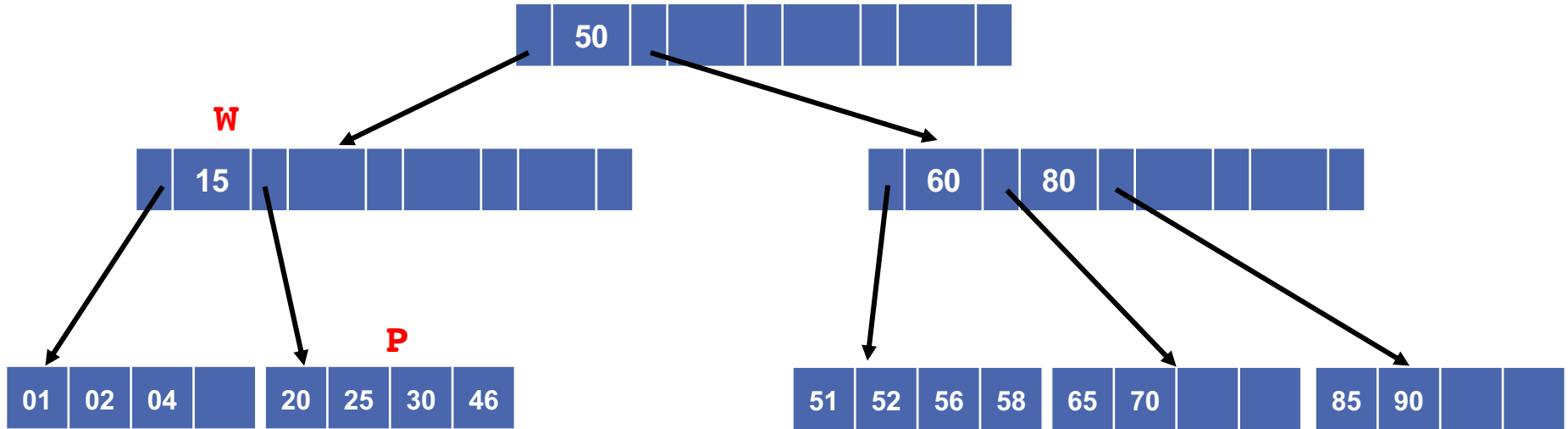
Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

Figura: Vanessa Braganholo (slides de árvore B)

Exemplo:

Exclusão da chave 40

Página **W** ficou com menos de **d** chaves necessário propagar operação



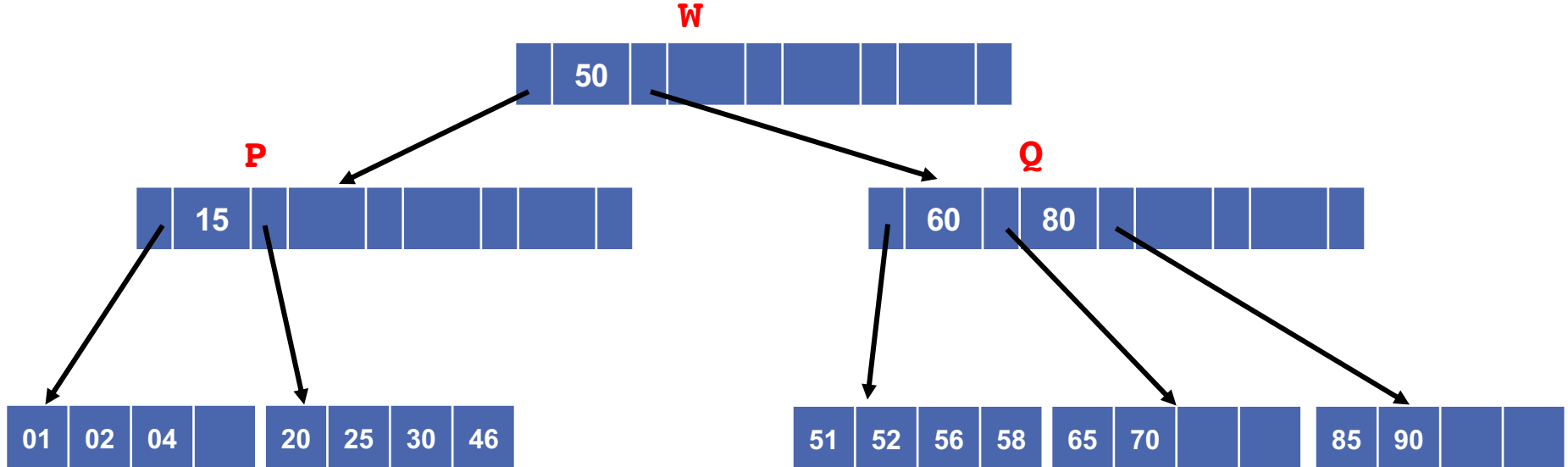
Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

Figura: Vanessa Braganholo (slides de árvore B)

Exemplo:

Exclusão da chave 40

Página **P** e **Q** são irmãs adjacentes
Soma de chaves de **P** e **Q** $< 2d$
CONCATENAR P e Q

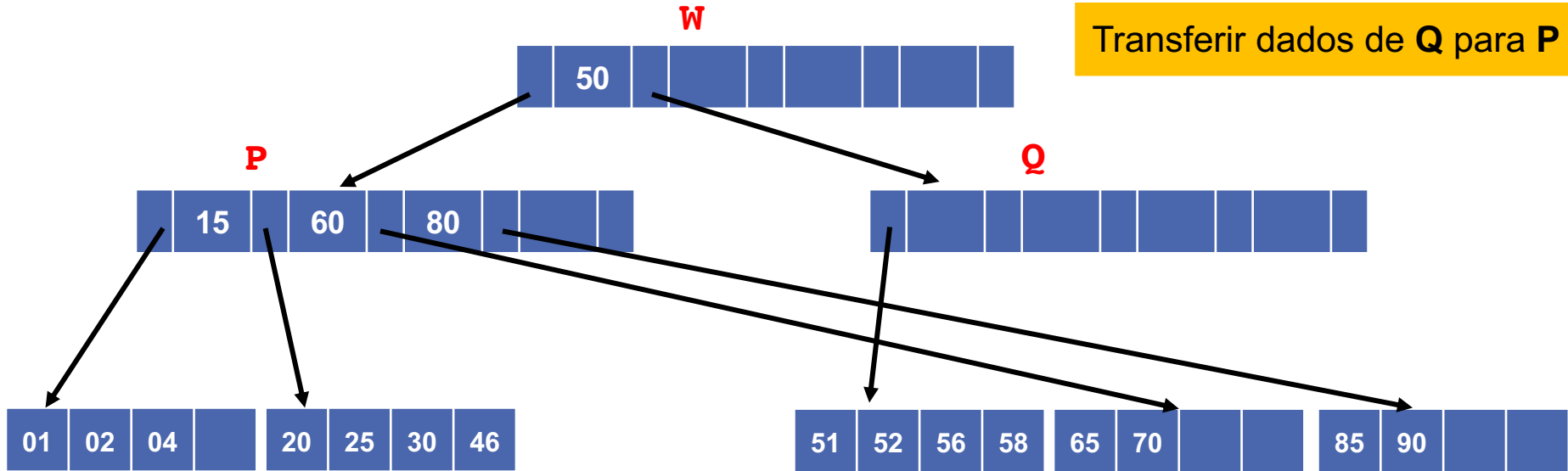


Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

Figura: Vanessa Braganholo (slides de árvore B)

Exemplo:

Exclusão da chave 40



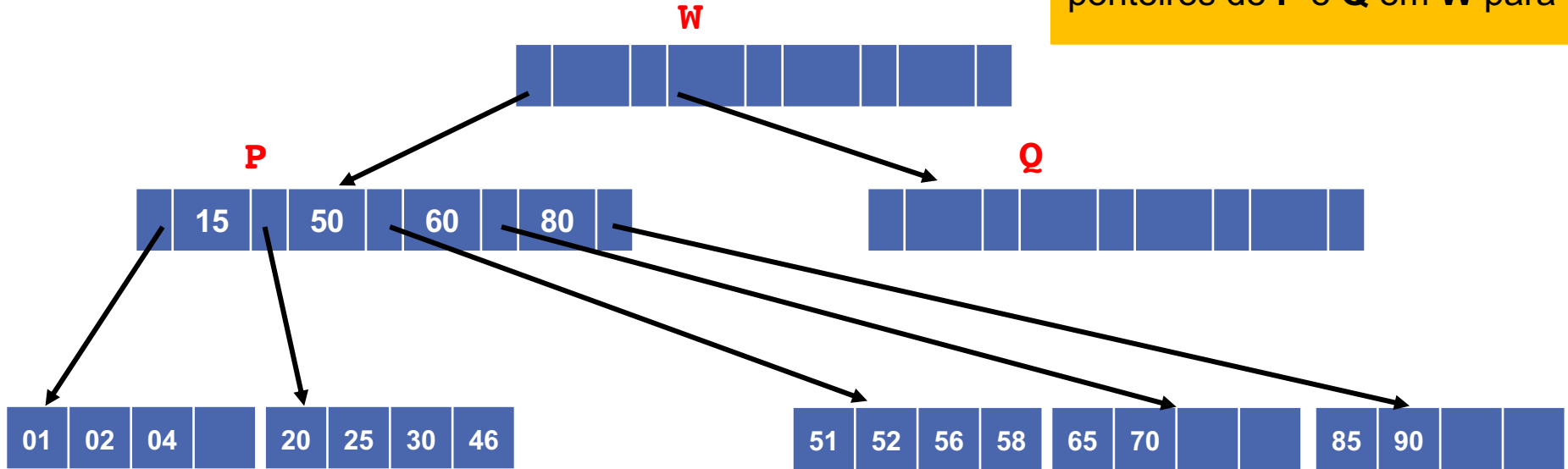
Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

Figura: Vanessa Braganholo (slides de árvore B)

Exemplo:

Exclusão da chave 40

Transferir chave que separa os ponteiros de **P** e **Q** em **W** para **P**



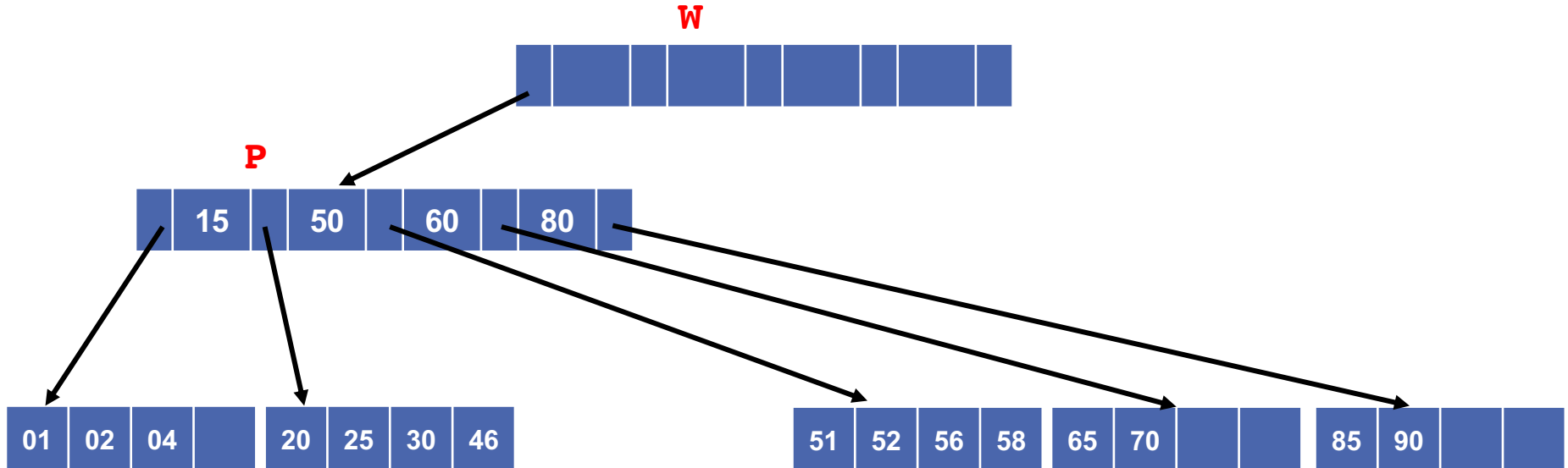
Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

Figura: Vanessa Braganholo (slides de árvore B)

Exemplo:

Exclusão da chave 40

Eliminar página **Q** e ponteiro



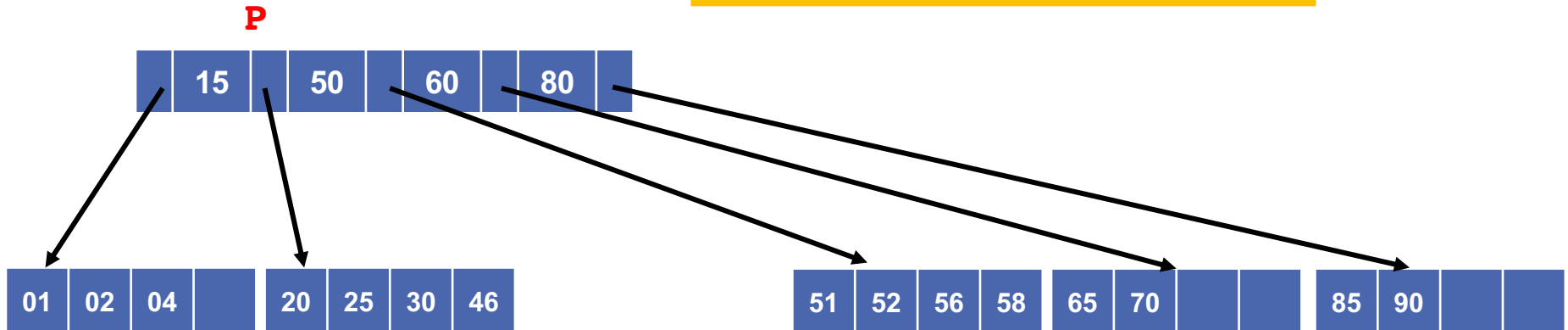
Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

Figura: Vanessa Braganholo (slides de árvore B)

Exemplo:

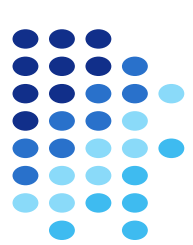
Exclusão da chave 40

W ficou vazia e era a raiz: eliminá-la
P passa a ser a nova raiz



Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

Figura: Vanessa Braganholo (slides de árvore B)

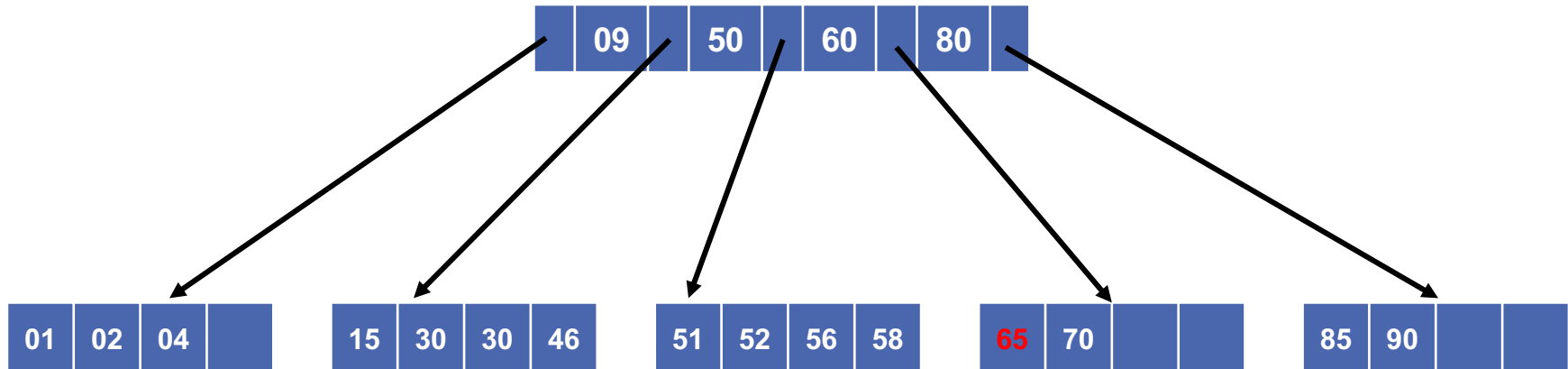


Redistribuição

- Ocorre quando a soma das entradas de **P** e de seu irmão adjacente **Q** é maior ou igual a **2d**
- Concatenar **P** e **Q**
 - Isso resulta em um nó **P** com mais de **2d** chaves, o que não é permitido
 - Particionar o nó concatenado, usando **Q** como novo nó
 - Essa operação não é propagável: o nó **W**, pai de **P** e **Q**, é alterado, mas seu número de chaves não é modificado

Exemplo:

Exclusão da chave 65



Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

Figura: Vanessa Braganholo (slides de árvore B)

Exemplo:

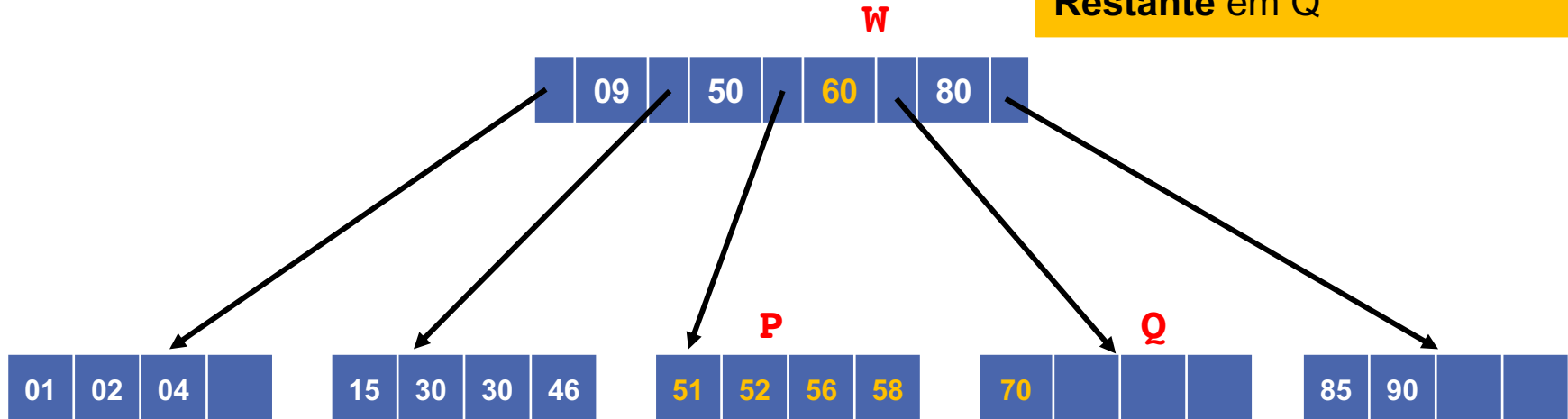
Exclusão da chave 65

Acomodar em **P** e **Q** as chaves:
51, 52, 56, 58, 60, 70

d chaves em **P**

chave **d+1** em **W**

Restante em **Q**



Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

Figura: Vanessa Braganholo (slides de árvore B)

Exemplo:

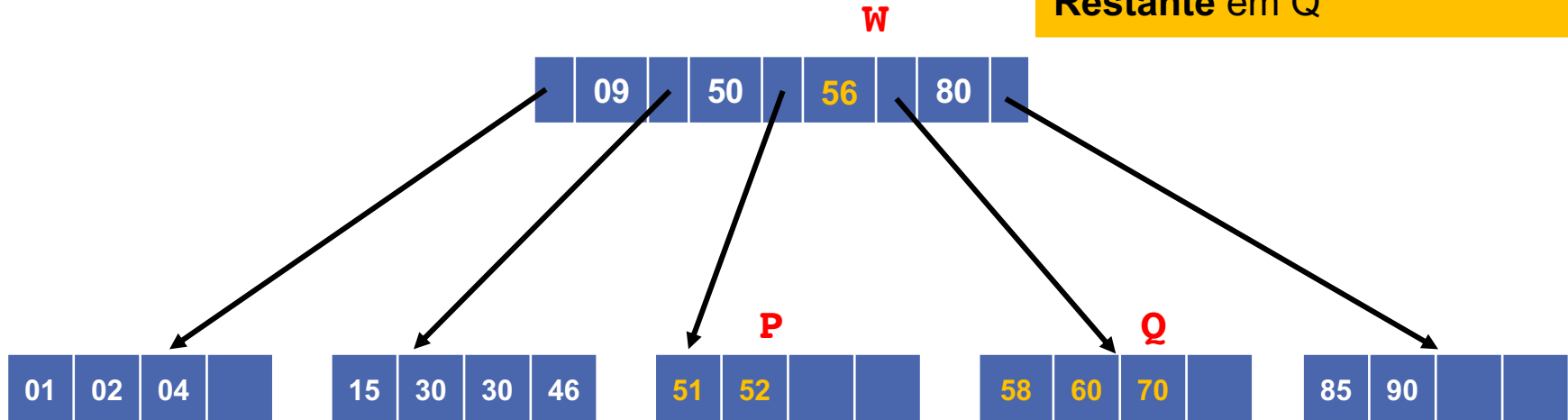
Exclusão da chave 65

Acomodar em **P** e **Q** as chaves:
51, 52, 56, 58, 60, 70

d chaves em **P**

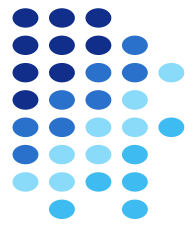
chave **d+1** em **W**

Restante em **Q**



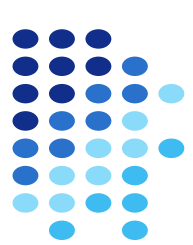
Atenção: os ponteiros dos nós folha foram omitidos por questões de legibilidade da figura. Na prática, todos apontam para NULL

Figura: Vanessa Braganholo (slides de árvore B)



E Quando as duas Alternativas são Possíveis?

- Quando for possível usar concatenação ou redistribuição (porque o nó possui 2 nós adjacentes, cada um levando a uma solução diferente), optar pela redistribuição
 - Ela é menos custosa, pois não se propaga
 - Ela evita que o nó fique cheio, deixando espaço para futuras inserções

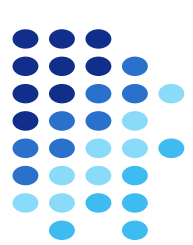


Exercício (parte 1)

- Desenhar uma **árvore B** de ordem 3 que contenha as seguintes chaves: **1, 3, 6, 8, 14, 32, 36, 38, 39, 41, 43**

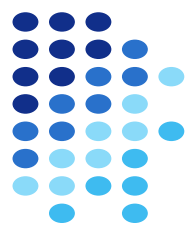
Dica: começar com uma **árvore B** vazia e ir inserindo uma chave após a outra

- Relembrando características de uma árvore B de ordem d
 - A raiz é uma folha ou tem no mínimo 2 filhos
 - Cada nó interno (não folha e não raiz) possui no mínimo $d + 1$ filhos
 - Cada nó tem no máximo $2d + 1$ filhos
 - Todas as folhas estão no mesmo nível



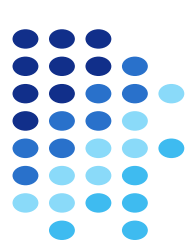
Exercício (parte 2)

- Sobre a árvore resultante do exercício anterior, realizar as seguintes operações:
 - a) Inserir as chaves **4, 5, 42, 2, 7**
 - b) Sobre o resultado do passo (a), excluir as chaves **14, 32**



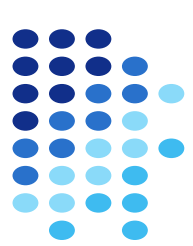
Busca de uma Chave X em uma Árvore B que Indexa Arquivo em Disco

1. Inicie lendo a raiz da árvore a partir do disco
2. Procure **x** dentro do nó lido (pode ser usada busca binária, pois as chaves estão ordenadas dentro do nó)
 - a) Se encontrou, encerra a busca;
 - b) Caso contrário, continue a busca, lendo o filho correspondente, a partir do disco
3. Continue a busca até que **x** tenha sido encontrado ou que a busca tenha sido feita em uma folha da árvore (retorna o último nó pesquisado – nó onde a chave está ou deveria estar)



Implementação

1. Um arquivo para guardar metadados, que contém
 - a) Um ponteiro para o nó raiz
 - b) Um ponteiro para o próximo nó livre do arquivo
2. Um arquivo para guardar os dados, estruturado em nós (ou páginas/blocos)

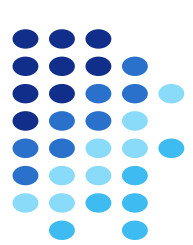


Implementação

- No arquivos de dados, cada nó possui
 - Inteiro representando o número de chaves (**m**) armazenadas no nó
 - Um ponteiro para o nó pai
 - Lista de **m+1** ponteiros para os nós filho
 - Lista de **m** registros

Considerações Sobre Implementação

- A cada vez que for necessário manipular um nó, ler o nó todo para a memória, e manipulá-lo em memória
- Depois, gravar o nó todo de volta no disco
- Na nossa disciplina, vamos simplificar:
 - da mesma forma que fazíamos uma função para ler um registro e gravar um registro inteiro, agora faremos uma função que lê uma página e grava uma página inteira no disco



Referências

- Material baseado nos slides de **Vanessa Braganholo**, Disciplina de Estruturas de Dados e Seus Algoritmos. Instituto de Computação. Universidade Federal Fluminense (UFF), Niterói, Brasil.
- Szwarcfiter, J.; Markezon, L. Estruturas de Dados e seus Algoritmos, 3a. ed. LTC. Cap. 10
- Inhaúma Neves Ferraz. Programação Com Arquivos. 2003. Editora: manole.
- Schildt, H. C Completo e Total. Ed. McGraw-Hill.