

# Estruturas de Dados Básicas II

Prof. Dr. Matheus da Silva Menezes



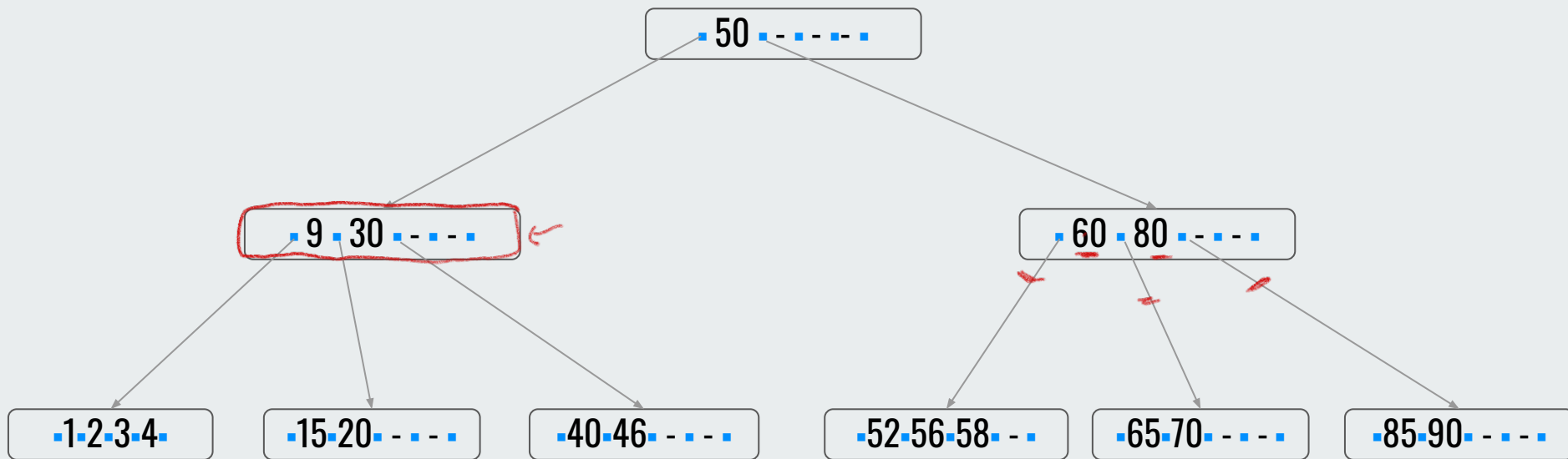
# Começando a Unidade 03.



## Estruturas de Dados Básicas II

ÁRVORE B  
Remoção

## Árvore B de ordem $d=2$



## Árvore B

### Características

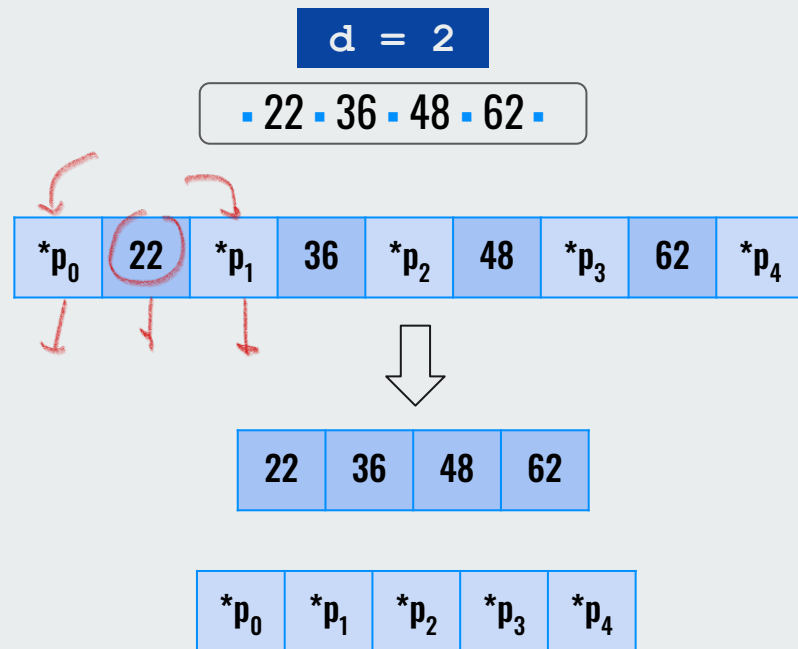
Considere  $d$  um número natural. Uma árvore **B** de ordem  $d$  é uma árvore que é vazia ou satisfaz as seguintes condições:

- Cada página possui entre  $d$  e  $2d$  chaves, exceto o nó raiz que possui entre  $1$  e  $2d$  chaves
- As chaves de cada página são ordenadas
- Cada nó diferente das folhas e da raiz tem, no mínimo,  $d + 1$  filhos
- Cada nó tem no máximo  $2d + 1$  filhos.

## Uma representação para nó de uma árvore B

### Exemplo C++

```
struct BTNode{  
    vector <int> chaves; //vetor de chaves  
    vector <BTNode*> filhos; //vetor de ponteiros para filhos  
    bool ehFolha; //indica se o nó é uma folha  
    int d; //Grau mínimo (capacidade da página)  
}
```



## Operações Básicas

1

**BUSCA**

2

**INSERÇÃO**

3

**REMOÇÃO**

## Operações Básicas

1 BUSCA

2 INSERÇÃO

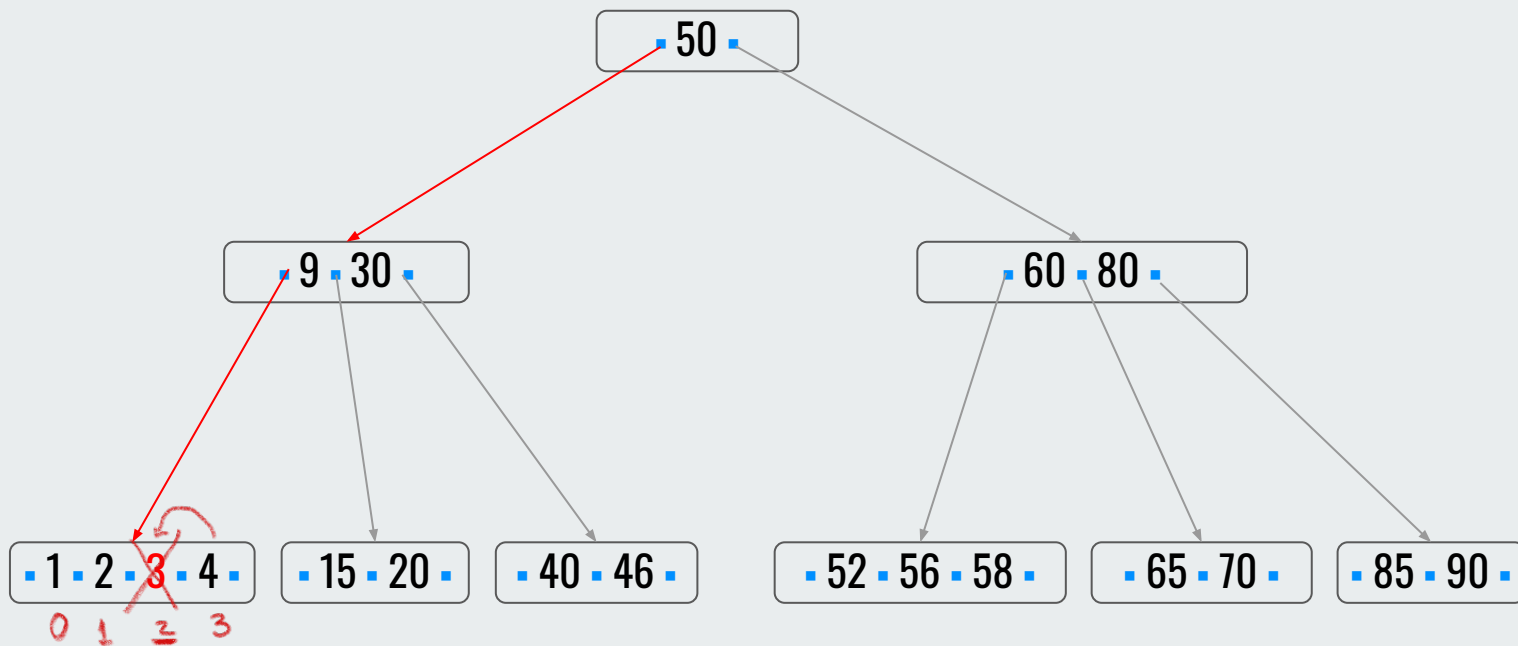
3 REMOÇÃO

Alguns casos a considerar...



## Caso 1: Remover uma chave de um nó folha que possui pelo menos $d+1$ elementos

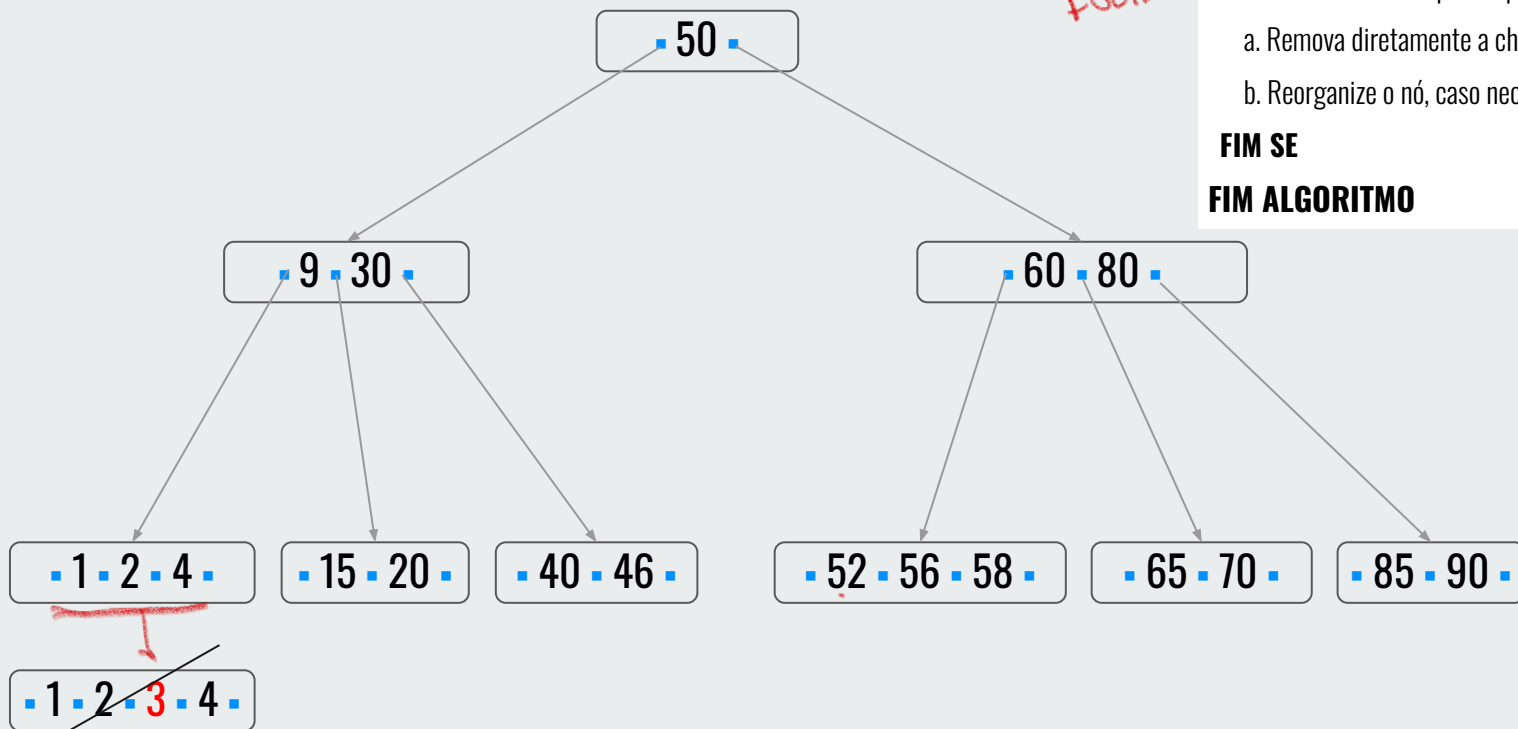
Excluir a chave: 3





## Caso 1: Remover uma chave de um nó folha que possui pelo menos $d+1$ elementos

Excluir a chave: 3



### Excluir\_no\_caso1

Localize a posição da chave no vetor de chaves do nó.

*FOLHA* = SE chave existe E o nó possui pelo menos  $d+1$  elementos ENTÃO

a. Remova diretamente a chave do nó.

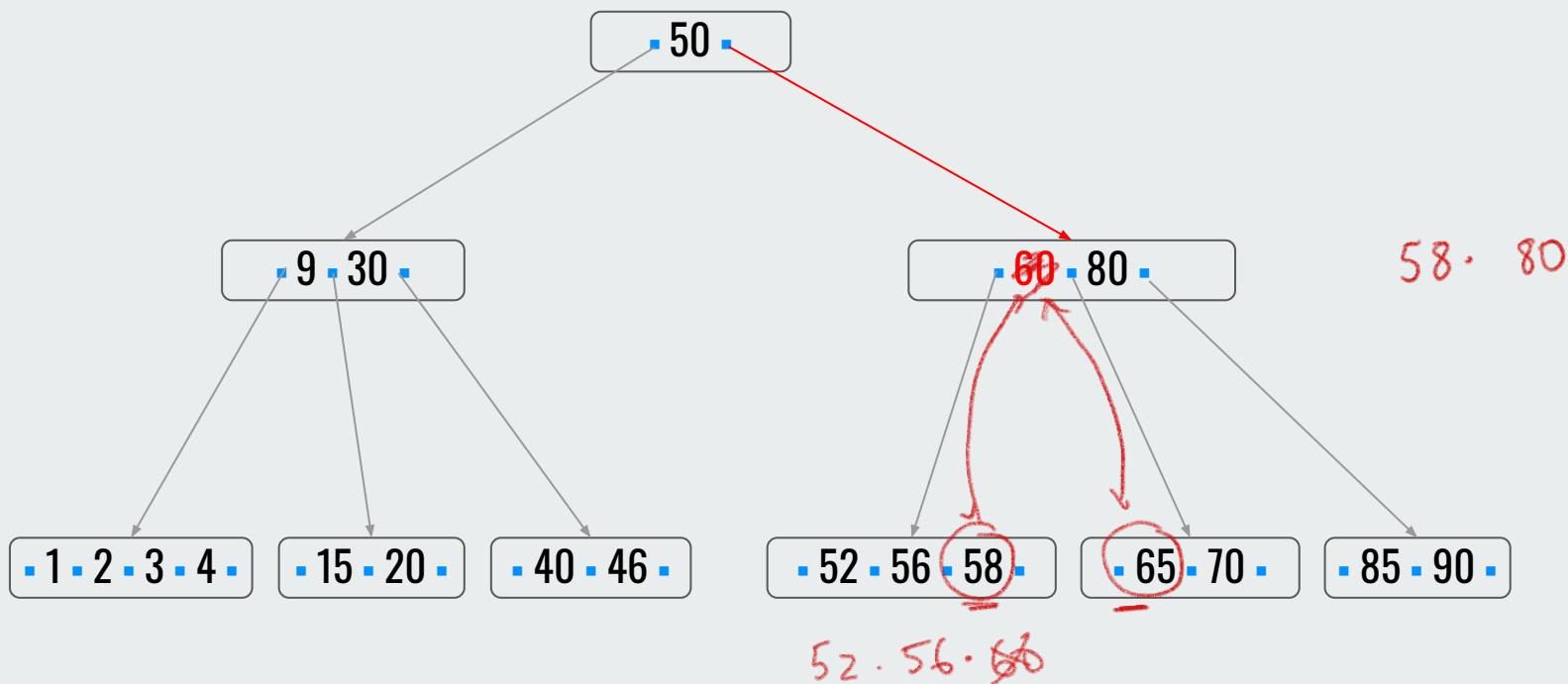
b. Reorganize o nó, caso necessário

FIM SE

FIM ALGORITMO

## Caso 2: Remover uma chave de um nó não folha

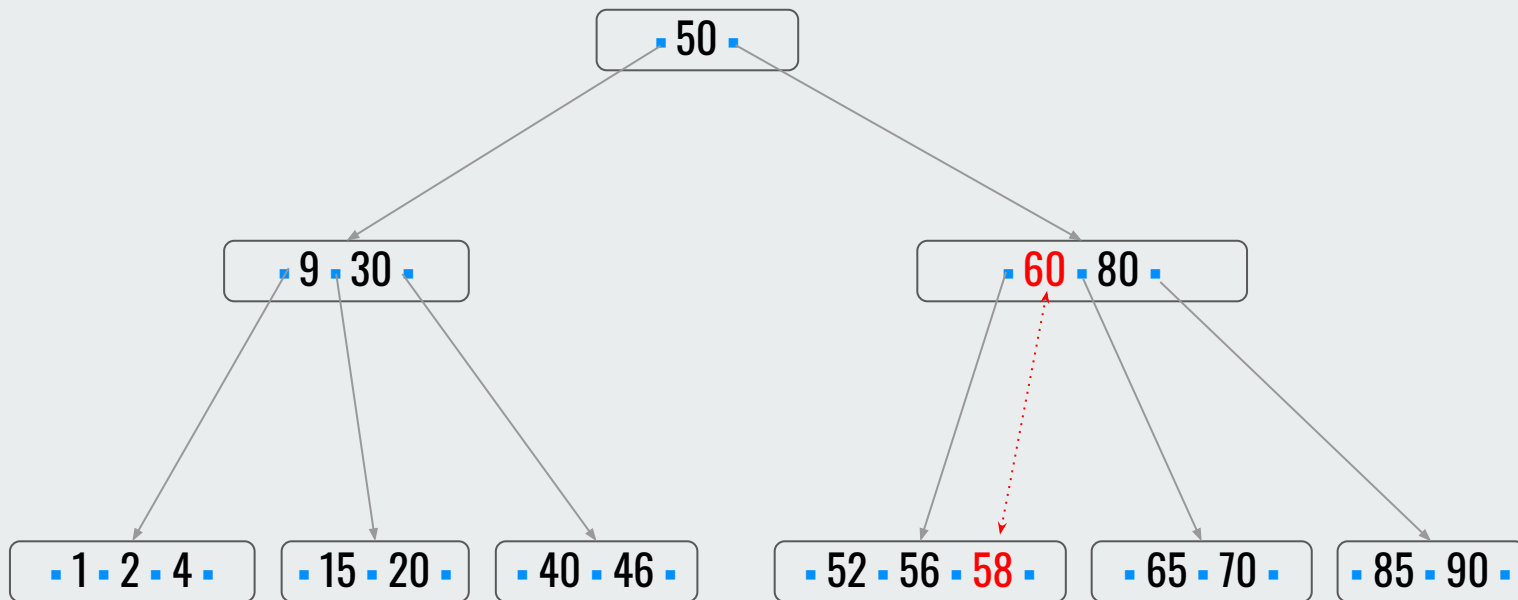
Excluir a chave: 60



## Caso 2: Remover uma chave de um nó não folha

Excluir a chave: 60

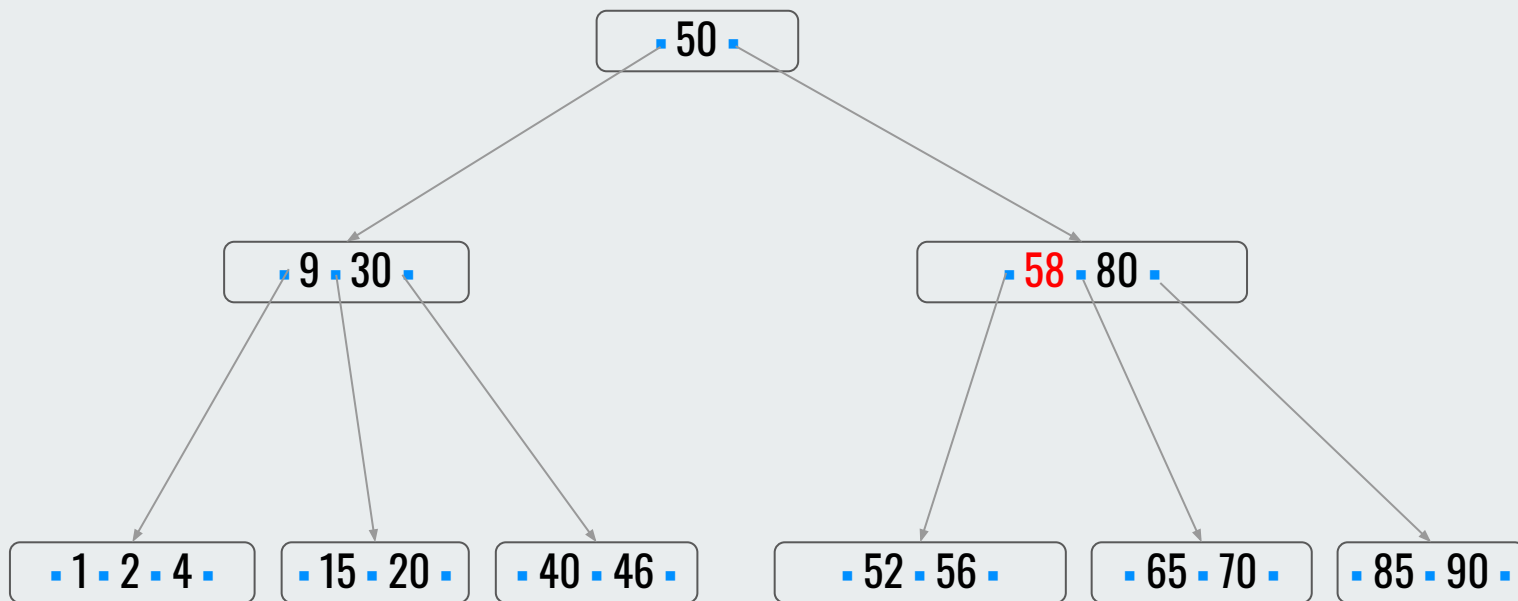
Substitua o valor a ser excluído pelo seu antecessor (maior elemento à esquerda) ou pelo sucessor (menor valor à direita)



## Caso 2: Remover uma chave de um nó não folha

Excluir a chave: 60

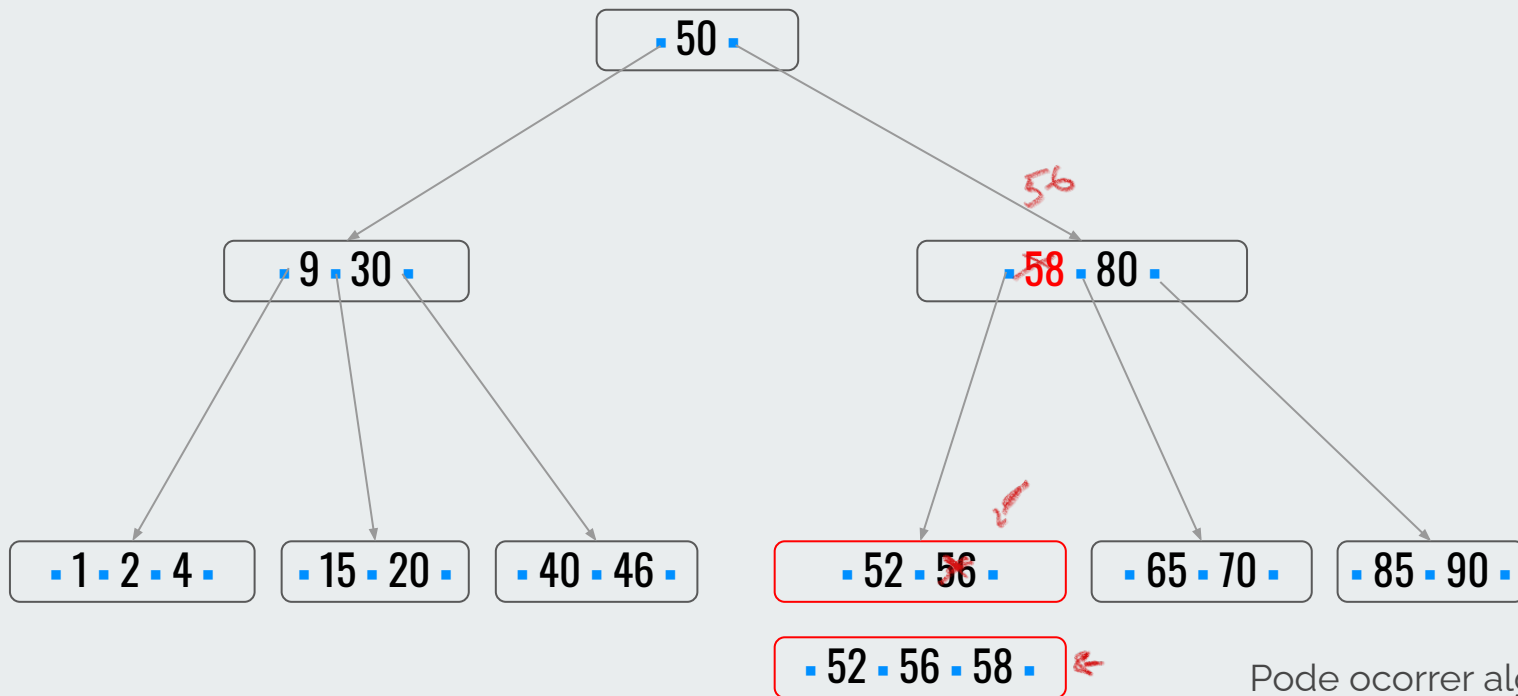
Substitua o valor a ser excluído pelo seu antecessor (maior elemento à esquerda) ou pelo sucessor (menor valor à direita)



## Caso 2: Remover uma chave de um nó não folha

Excluir a chave: 60

Substitua o valor a ser excluído pelo seu antecessor (maior elemento à esquerda) ou pelo sucessor (menor valor à direita)



ANTES → 3 Elementos    DEPOIS → 2 Elementos

Pode ocorrer algum problema? ✓

## Árvore B

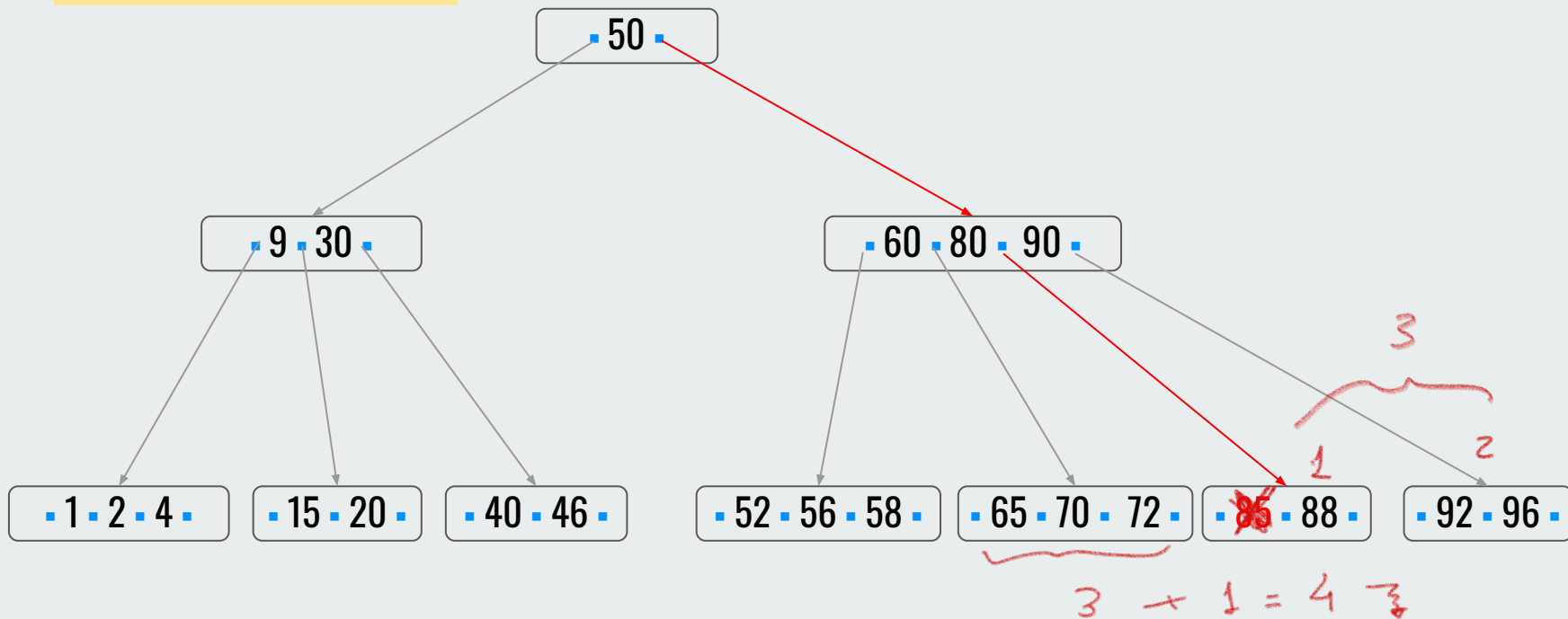
### Características

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

- Cada página possui entre  $d$  e  $2d$  chaves, exceto o nó raiz que possui entre  $1$  e  $2d$  chaves
- As chaves de cada página são ordenadas
- Cada nó diferente das folhas e da raiz tem, no mínimo,  $d + 1$  filhos
- Cada nó tem no máximo  $2d + 1$  filhos.

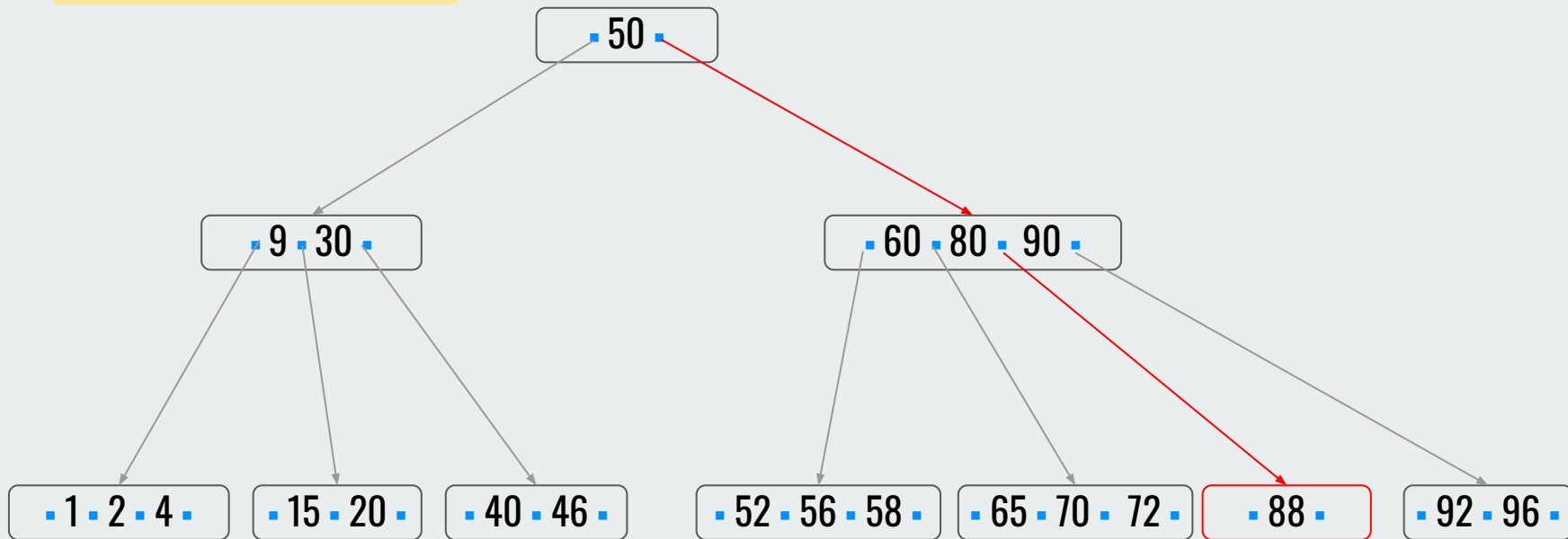
**Caso 3:** Remover uma chave de uma folha com  $d$  elementos, mas que possui uma página irmã adjacente e que juntas possuem menos de  $2d$  elementos

Excluir a chave: **85**



**Caso 3:** Remover uma chave de uma folha com  $d$  elementos, mas que possui uma página irmã adjacente e que juntas possuem menos de  $2d$  elementos

Excluir a chave: **85**



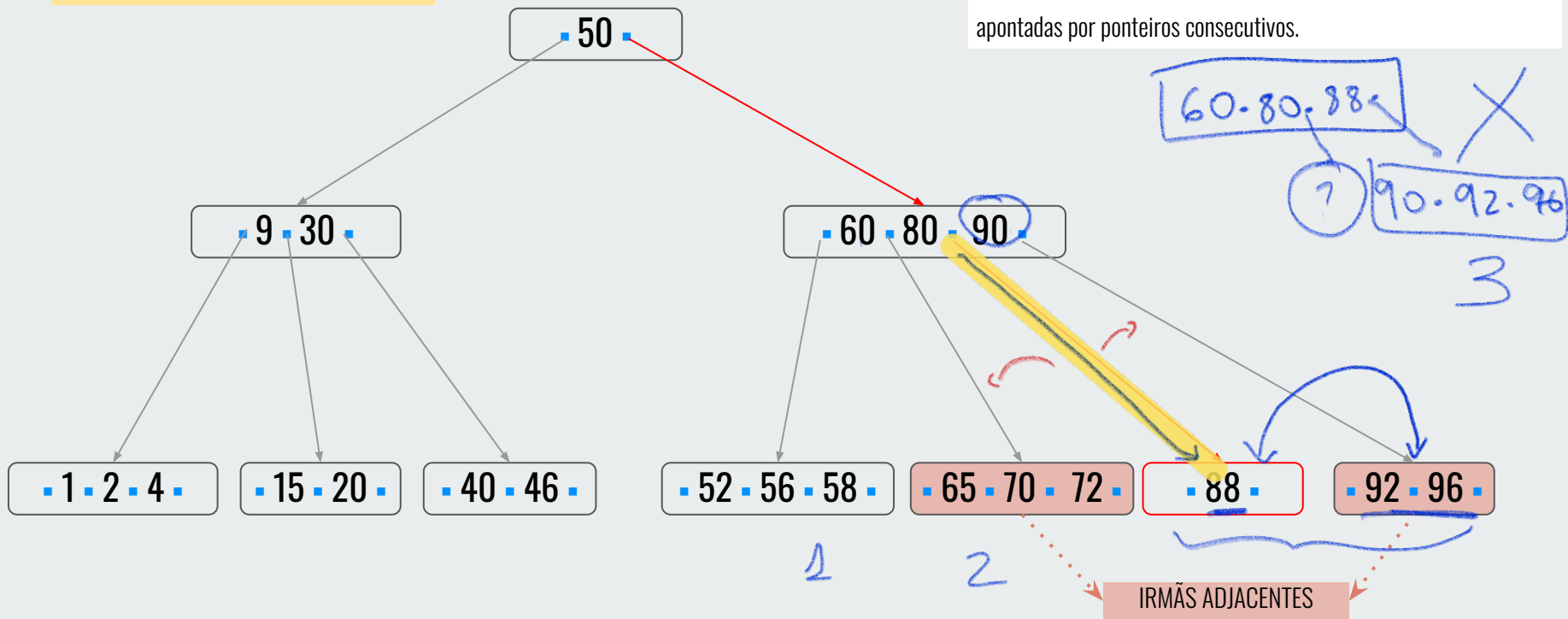
PROBLEMA: A página deve ter no mínimo  $d$  elementos



**Caso 3:** Remover uma chave de uma folha com  $d$  elementos, mas que possui uma página irmã adjacente e que juntas possuem menos de  $2d$  elementos

Excluir a chave: 85

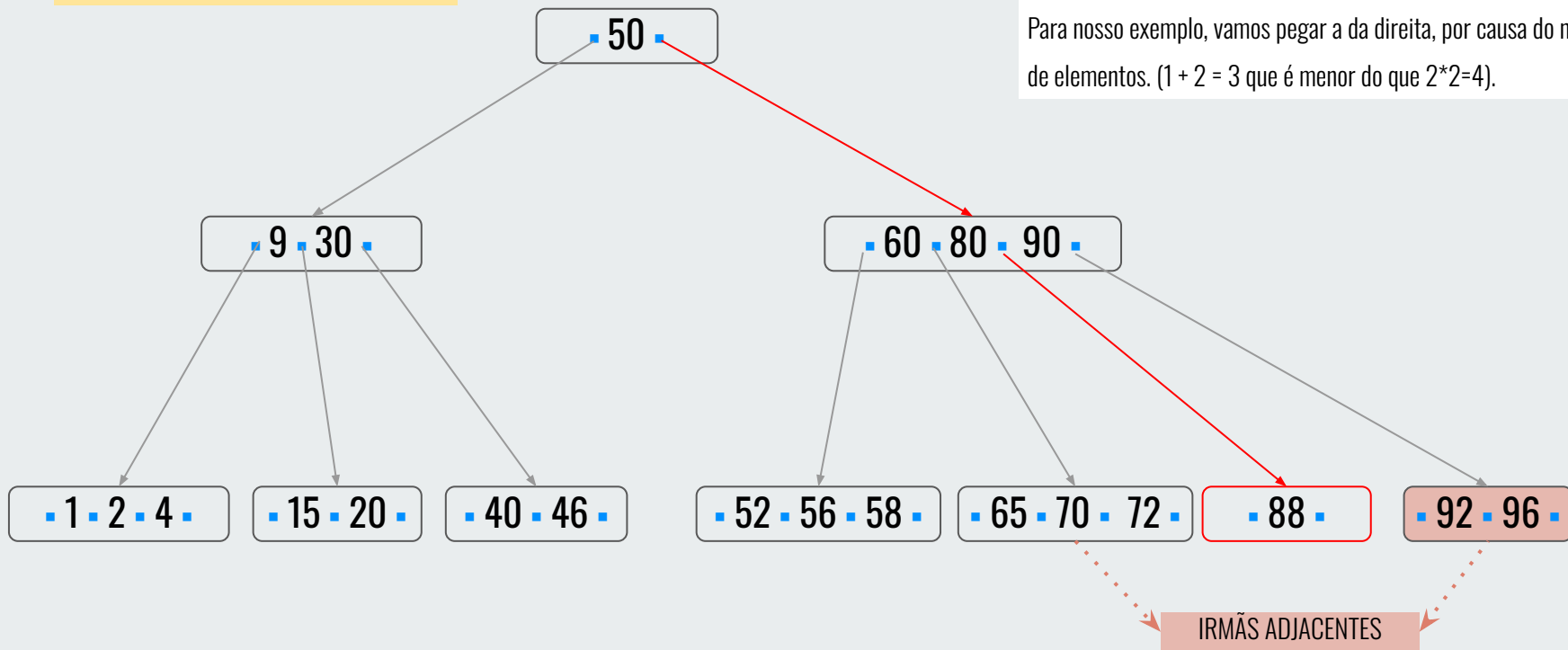
**P** e **Q** são irmãs adjacentes se possuem o mesmo pai **W** e são apontadas por ponteiros consecutivos.



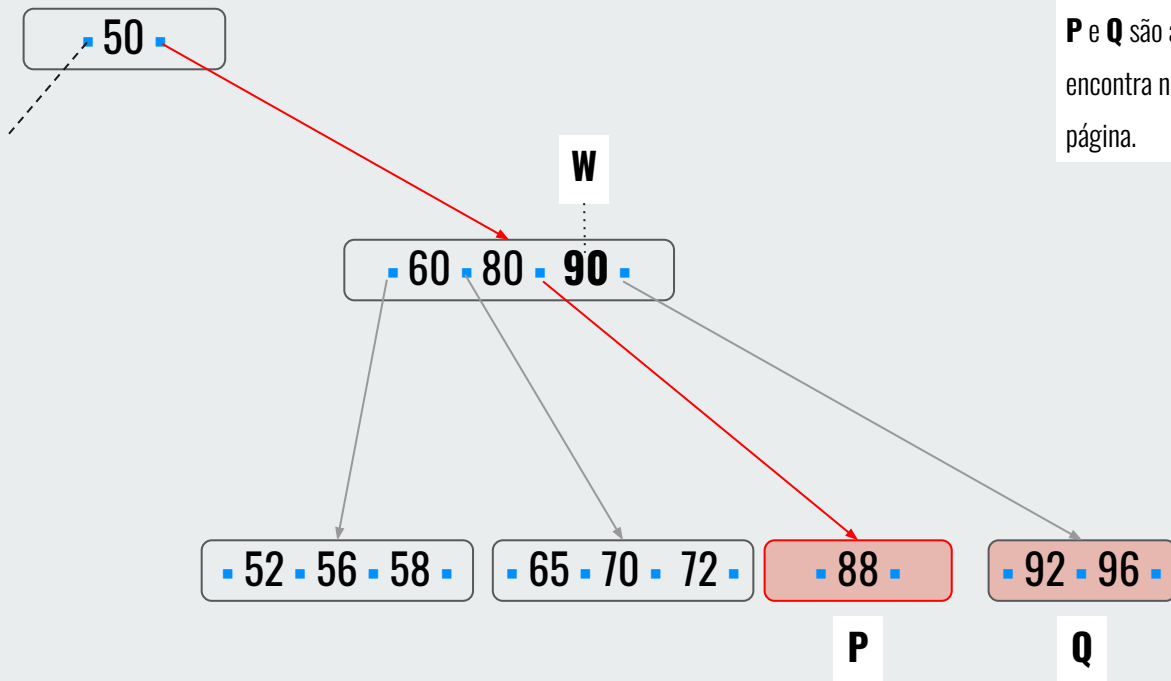
**Caso 3:** Remover uma chave de uma folha com  $d$  elementos, mas que possui uma página irmã adjacente e que juntas possuem menos de  $2d$  elementos

Excluir a chave: **85**

A escolha de qual irmã adjacente vai depender da implementação.  
Para nosso exemplo, vamos pegar a da direita, por causa do número de elementos. ( $1 + 2 = 3$  que é menor do que  $2 \cdot 2 = 4$ ).



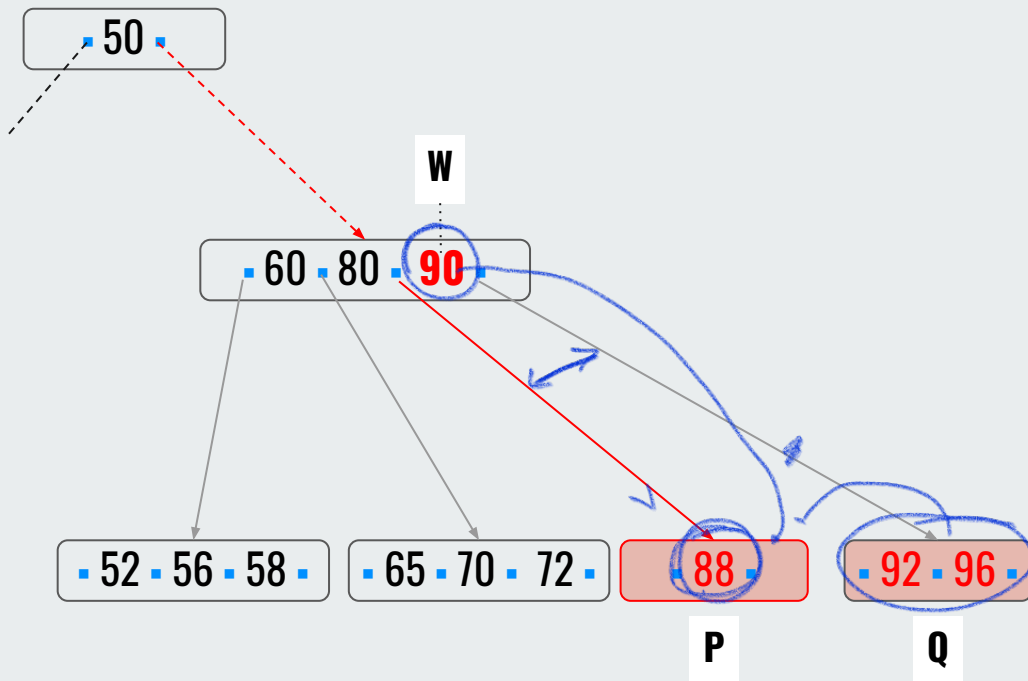
**Caso 3:** Remover uma chave de uma folha com  $d$  elementos, mas que possui uma página irmã adjacente e que juntas possuem menos de  $2d$  elementos



CONCATENAÇÃO:

**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.

**Caso 3:** Remover uma chave de uma folha com  $d$  elementos, mas que possui uma página irmã adjacente e que juntas possuem menos de  $2d$  elementos



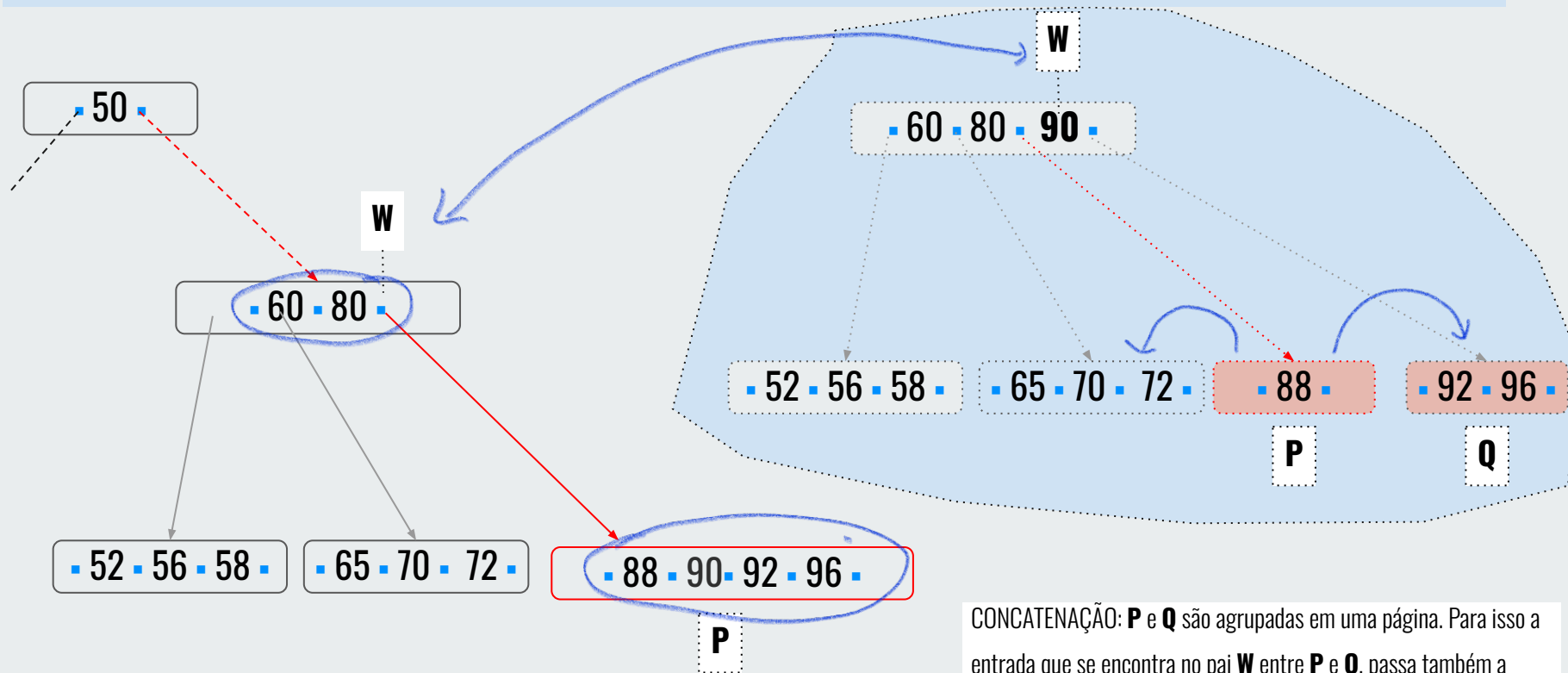
CONCATENAÇÃO:

**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.

?

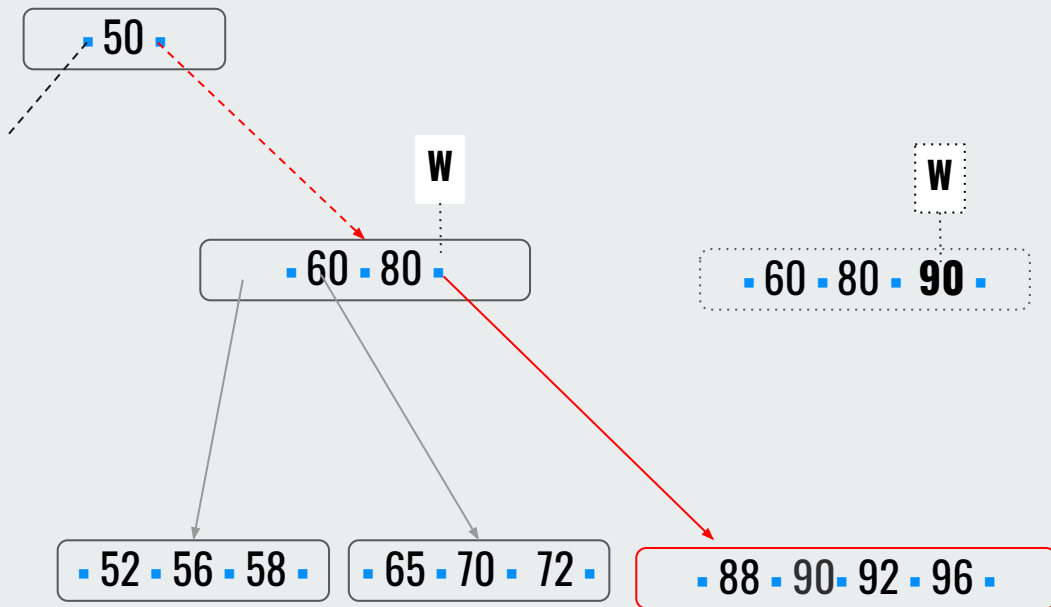
**88** **90** **92** **96**

**Caso 3:** Remover uma chave de uma folha com  $d$  elementos, mas que possui uma página irmã adjacente e que juntas possuem menos de  $2d$  elementos



CONCATENAÇÃO: **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.

**Caso 3:** Remover uma chave de uma folha com  $d$  elementos, mas que possui uma página irmã adjacente e que juntas possuem menos de  $2d$  elementos



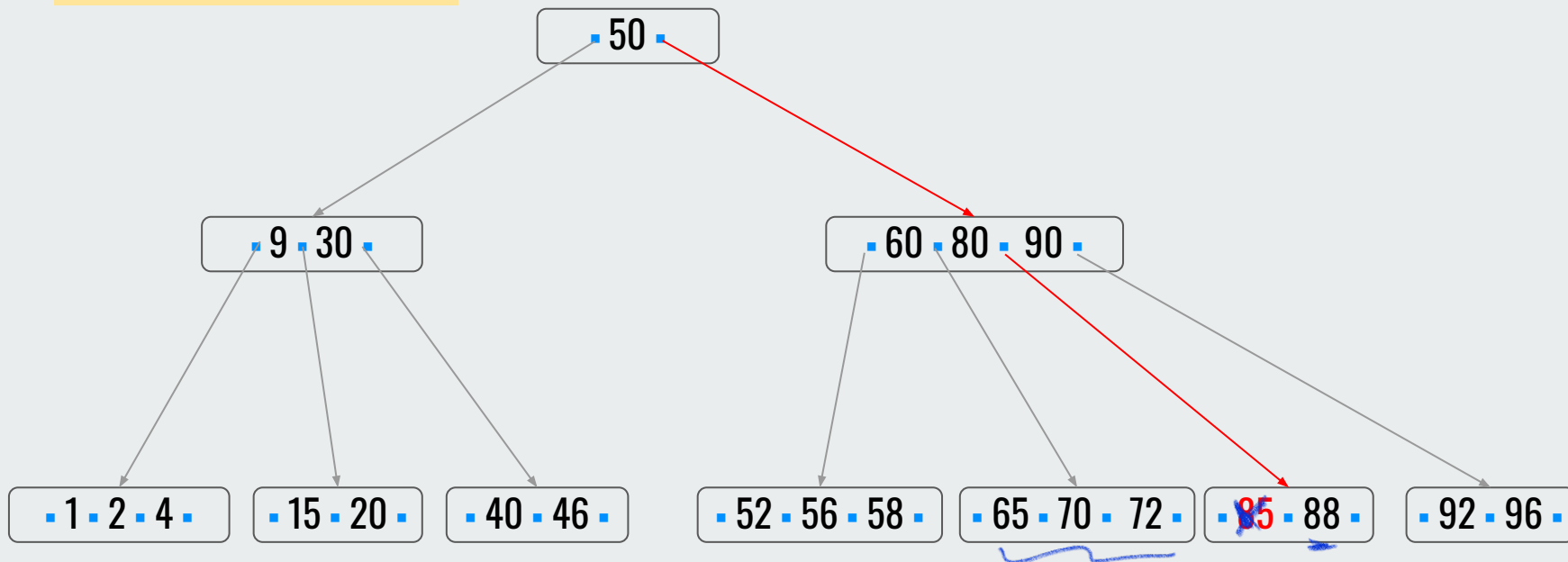
CONCATENAÇÃO: **P** e **Q** são agrupadas em uma página. Para isso a entrada que se encontra no pai **W** entre **P** e **Q**, passa também a fazer parte da nova página.

**RESULTADO:** W “perdeu” um elemento

Pode ocorrer algum problema?

**Caso 4:** Remover uma chave de uma folha com  $d$  elementos, mas que possui uma página irmã adjacente e que juntas possuem  $2d$  ou mais elementos.

Excluir a chave: **85**







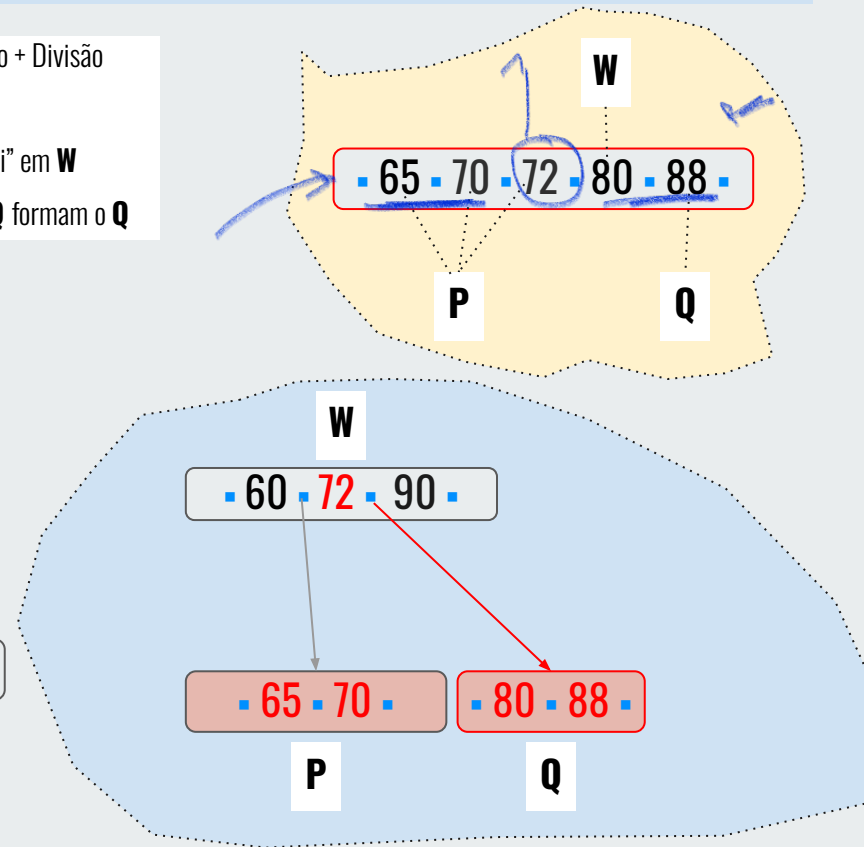
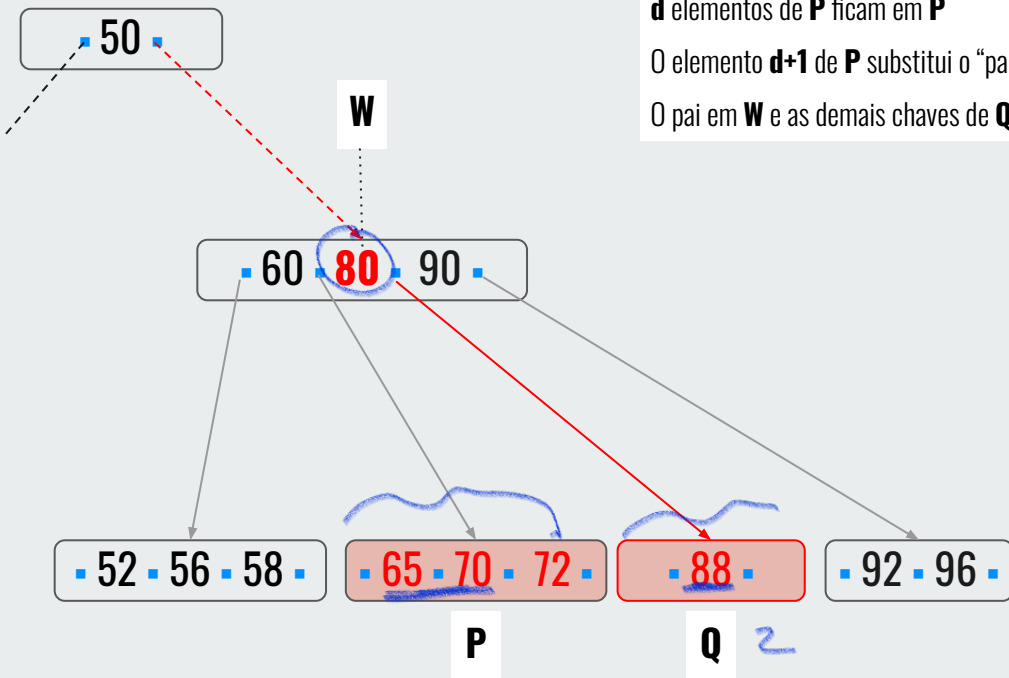
**Caso 4:** Remover uma chave de uma folha com  $d$  elementos, mas que possui uma página irmã adjacente e que juntas possuem  $2d$  ou mais elementos.

**REDISTRIBUIÇÃO** = Concatenação + Divisão

$d$  elementos de **P** ficam em **P**

O elemento  $d+1$  de **P** substitui o “pai” em **W**

O pai em **W** e as demais chaves de **Q** formam o **Q**



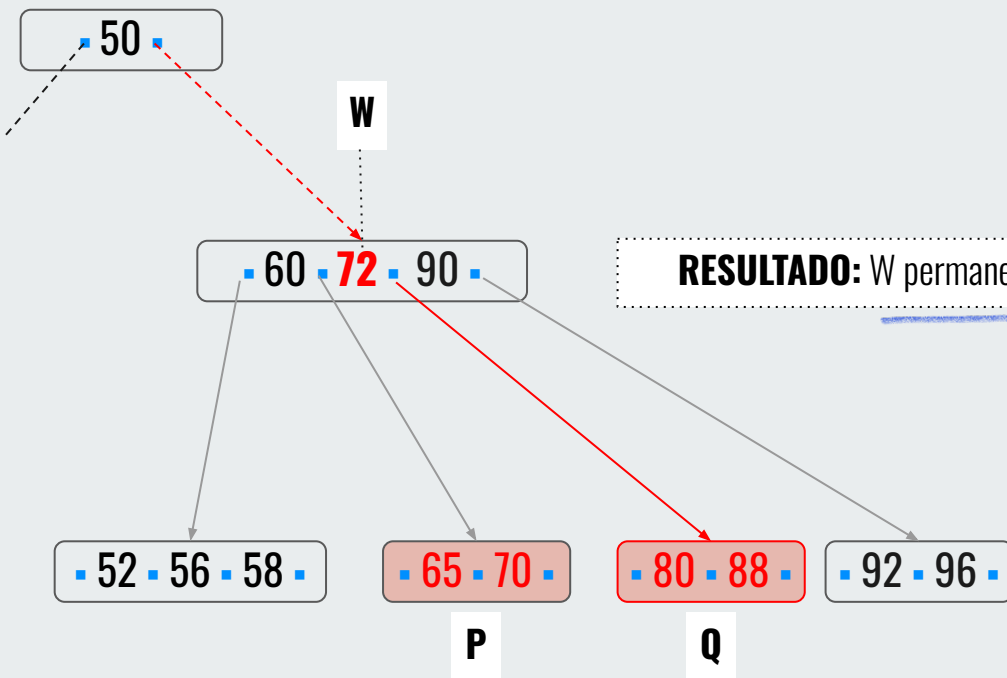
**Caso 4:** Remover uma chave de uma folha com  $d$  elementos, mas que possui uma página irmã adjacente e que juntas possuem  $2d$  ou mais elementos.

**REDISTRIBUIÇÃO** = Concatenação + Divisão

$d$  elementos de **P** ficam em **P**

O elemento  $d+1$  de **P** substitui o “pai” em **W**

O pai em **W** e as demais chaves de **Q** formam o **Q**



**RESULTADO:** W permanece do mesmo tamanho

Pode ocorrer algum problema?

**RESULTADO:** P e Q ainda possuem espaço para novas inserções

## CONCATENAÇÃO OU REDISTRIBUIÇÃO?

**Concatenação** pode se propagar (RECURSIVA)

Nó (página) resultante fica cheio, dificultando futuras inserções.

Redistribuição não é propagável.

Sobra espaço nos nós para futuras inserções.

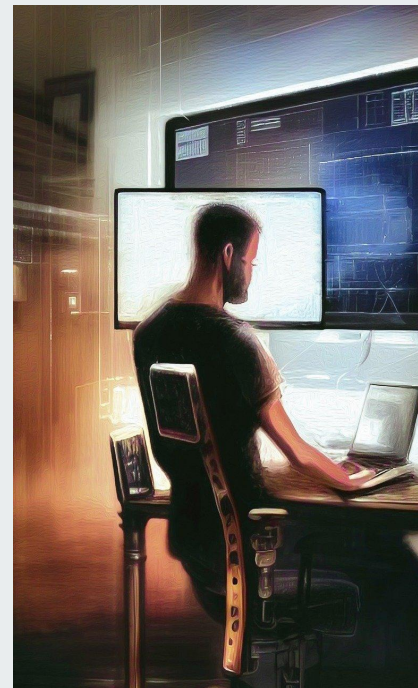
Se puder, escolha esse!



# Vamos programar?



1. Implemente as seguintes funções em Árvore B
  - a. Remoção



# Dúvidas?!

