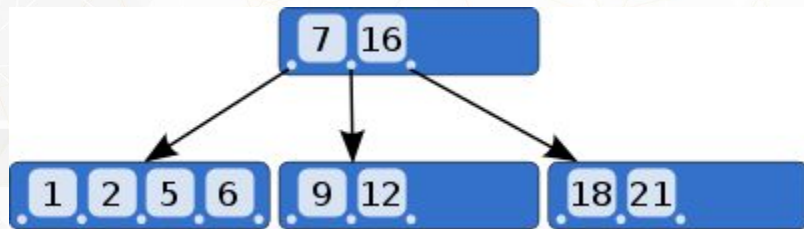


# Árvore B



Lucas Samuel  
Olivelton J Boelter  
Ricardo M. de Oliveira



# Árvore B

**O que é uma árvore B ou árvore b-tree?**

- **São árvores de estrutura de dados auto-balanceadas.**
- **Permite acesso, inserção e remoção em tempo logarítmico.**

# Árvore B

## Complexibilidade de tempo

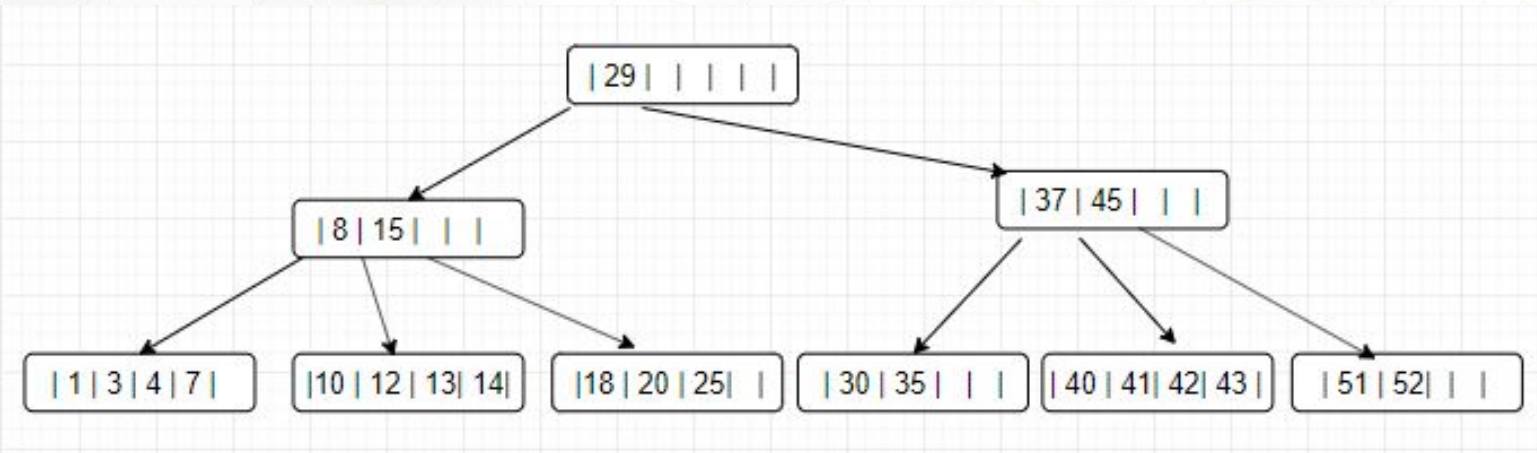
| Algoritmo | Caso Médio  | Pior Caso   |
|-----------|-------------|-------------|
| Espaço    | $O(n)$      | $O(n)$      |
| Busca     | $O(\log n)$ | $O(\log n)$ |
| Inserção  | $O(\log n)$ | $O(\log n)$ |
| Remoção   | $O(\log n)$ | $O(\log n)$ |



# Ordem da Árvore

- Número mínimo de elementos que cada página (exceto raiz) pode ter (Cormen, 2001; Bayer e McCreight, 1972)
- Número de filhos que cada página pode ter (Knuth, 1978)

# Ordem da Árvore



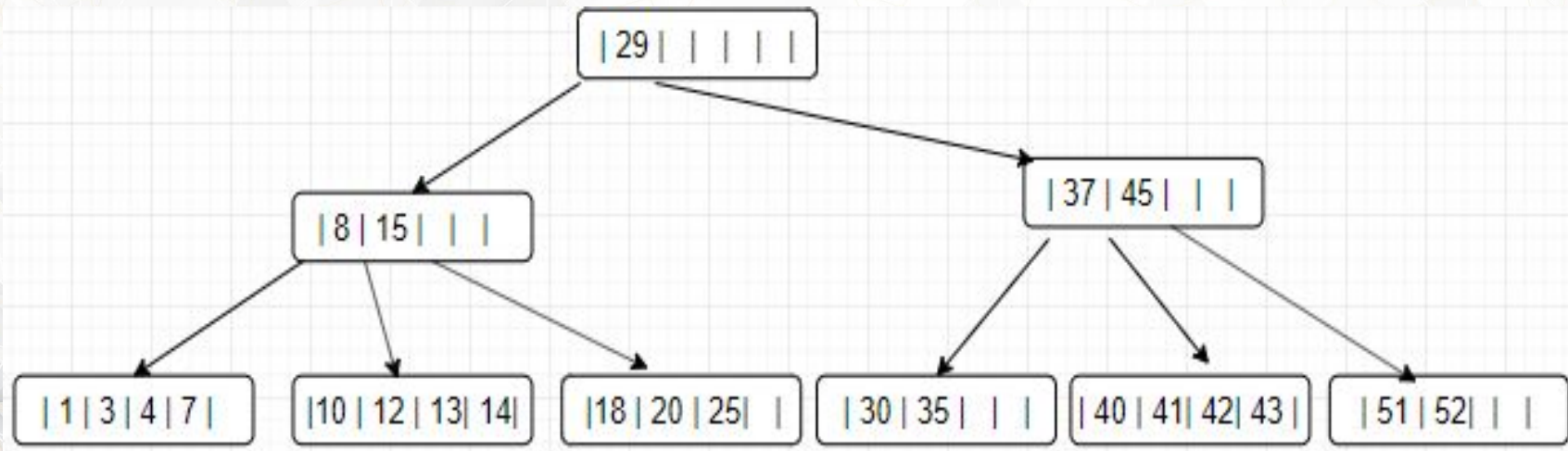
# Regras da árvore B

- Cada página deve ter pelo menos 50% ocupação (Exceto a raiz).
- O número de filhos (Exceto folha), deve ser o número de chaves +1
- Todas as folhas estão no mesmo nível.



# Busca em árvore B

- Exemplo - Localizar chave 18



# Inserção

## *1ª abordagem*

É necessário se criar um nó vazio, e inserir o primeiro elemento.

Importe lembrar que a inserção de baixo para cima (Bottom-up) podem ocorrer em duas situações

1. Folhas com número menor de chaves que o máximo permite.
2. Folhas completas ou número máximo permitido

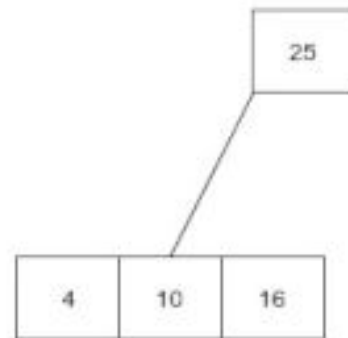


# Inserção

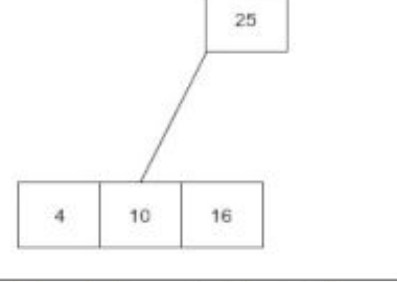
## *2ª abordagem*

A outra forma de inserção é a de cima para baixo (Top-down).

1. Porém quando inserimos um valor na página que está cheia é necessário realizar um split.
2. E esta abordagem previne excesso de processamento.



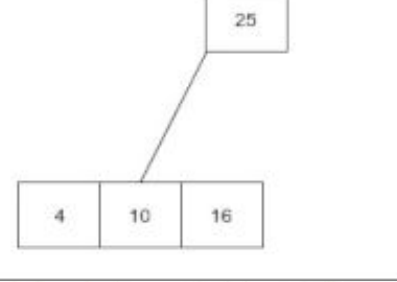
# Split



**A função do split é dividir o nó em duas partes;**

- Subir o valor central do nó para um nó acima.
- Caso seja a raiz, é criada uma nova raiz.

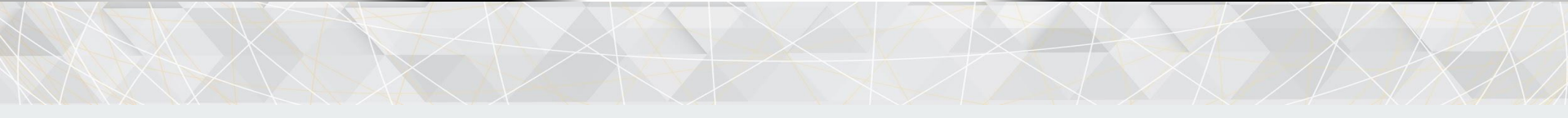
# Split



**Para realizar o split temos que fazer as seguinte:**

- Teremos que calcular a mediana do valores do nó
- Verifica se quem sofreu o split não seja a raiz.
- Caso seja a raiz, é criada uma nova raiz.





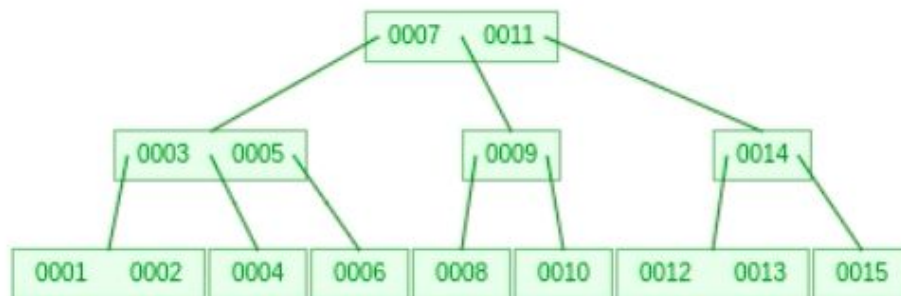
# Estrutura da página

|   |    |         |    |         |     |      |           |    |
|---|----|---------|----|---------|-----|------|-----------|----|
| N | P0 | C0 - D0 | P1 | C1 - D1 | ... | Pn-1 | Cn-1 Dn-1 | Pn |
|---|----|---------|----|---------|-----|------|-----------|----|

- N = Número de elemento presentes na página
- $C_i$  = Chave do registro (código)
- $D_i$  = Dados
- $P_i$  = Ponteiro para o i-ésimo filho

- ☒ Max. Degree = 3 ☐ Preemptive Split / Merge (Even)
- ☐ Max. Degree = 4
- ☐ Max. Degree = 5
- ☐ Max. Degree = 6
- ☐ Max. Degree = 7

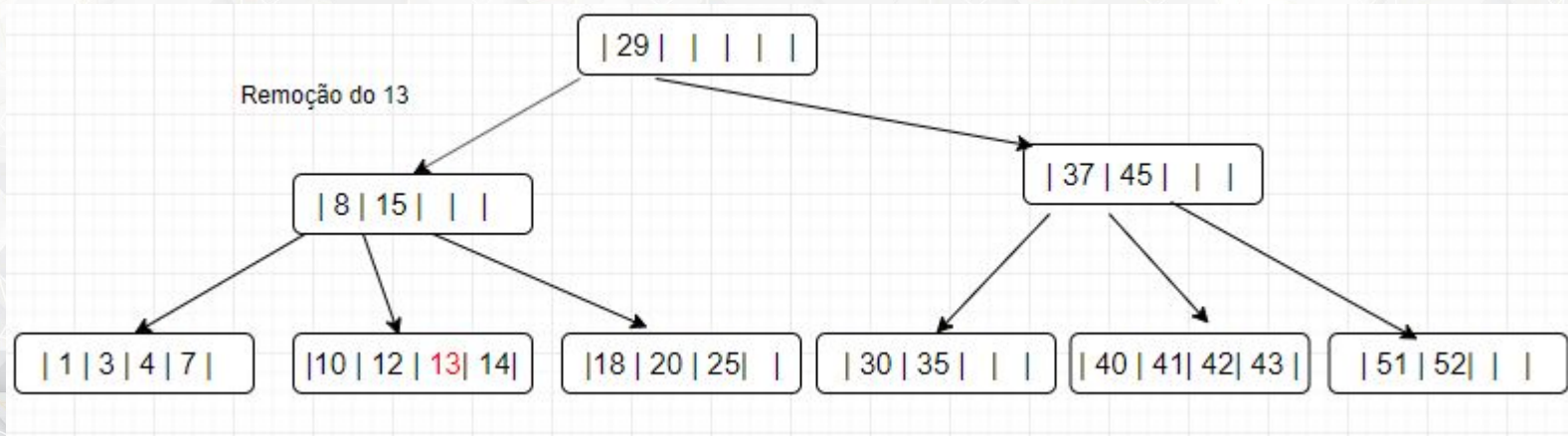
Element 0002 found





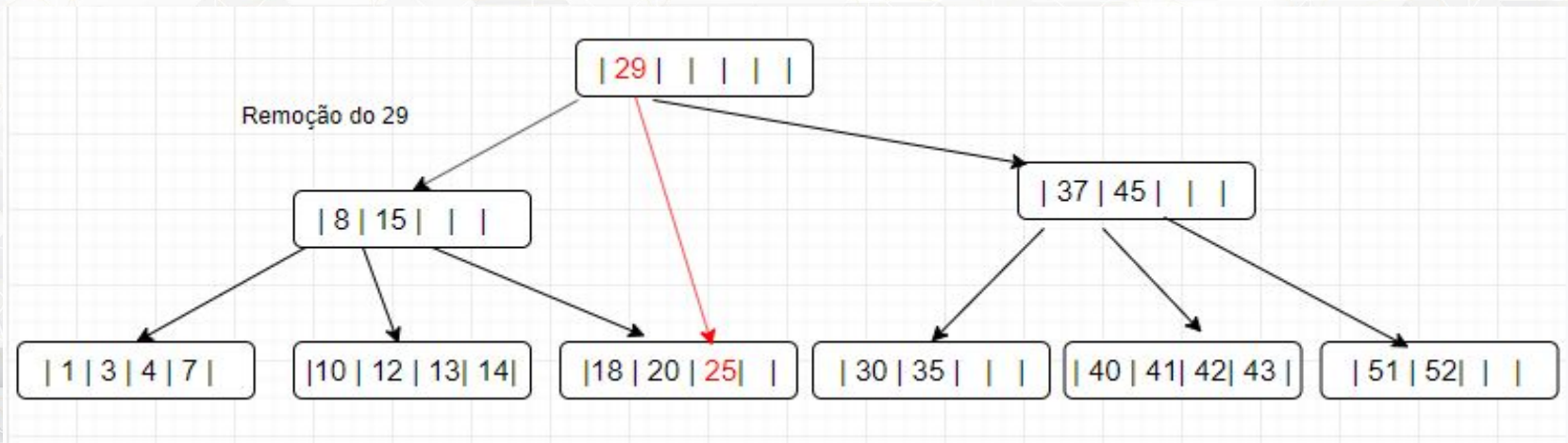
# Remoção em árvore B

- Caso 1: se o elemento estiver em uma folha e a folha mantiver 50% de ocupação.



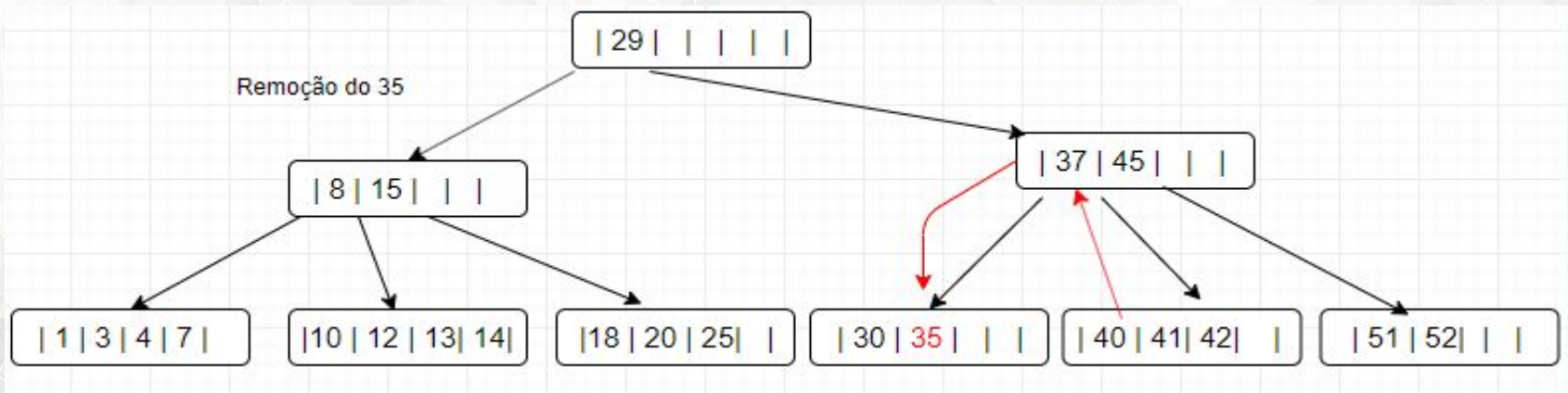
# Remoção em árvore B

- Caso 2: se o elemento Não estiver em uma folha, trocá-lo pelo seu antecessor



# Remoção em árvore B

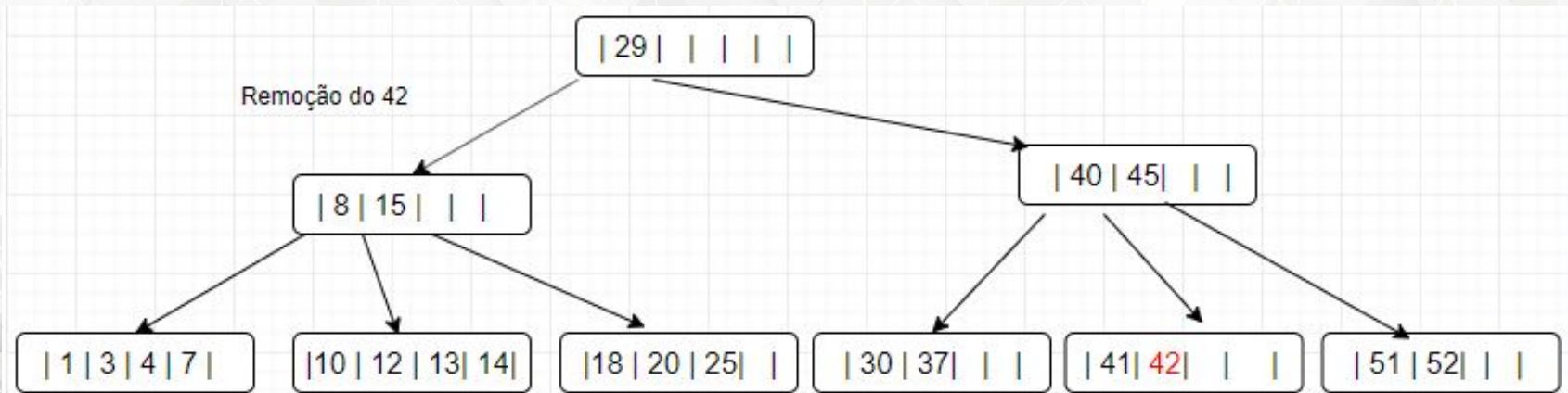
- Caso 3: se a folha ficar com menos de 50% de ocupação, mas a página irmã puder ceder uma chave;





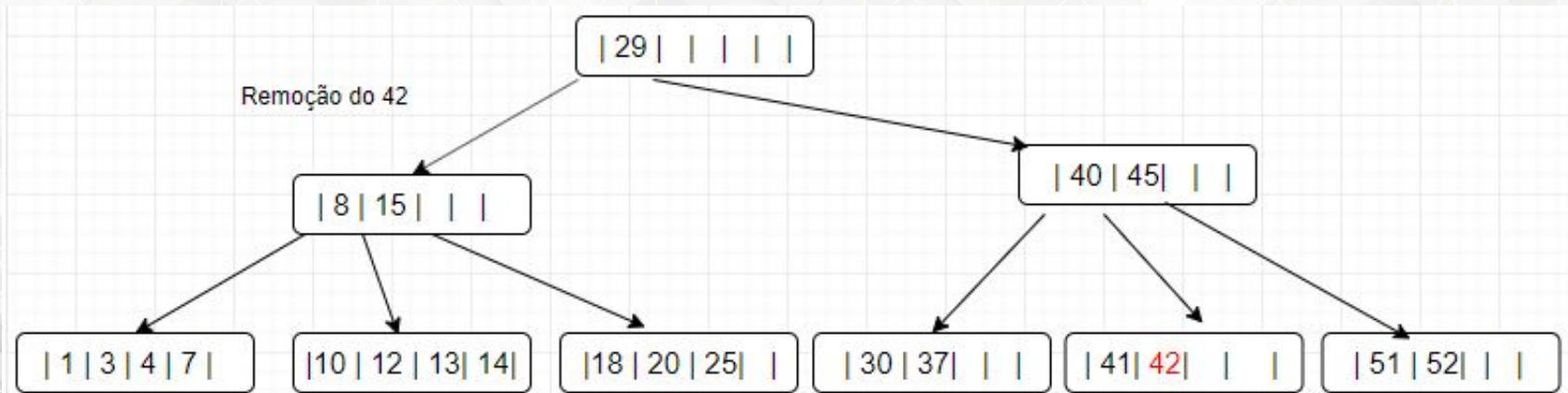
# Remoção em árvore B

- Caso 4: se a folha ficar com menos de 50% de ocupação, e as páginas irmãs não puderem ceder uma chave

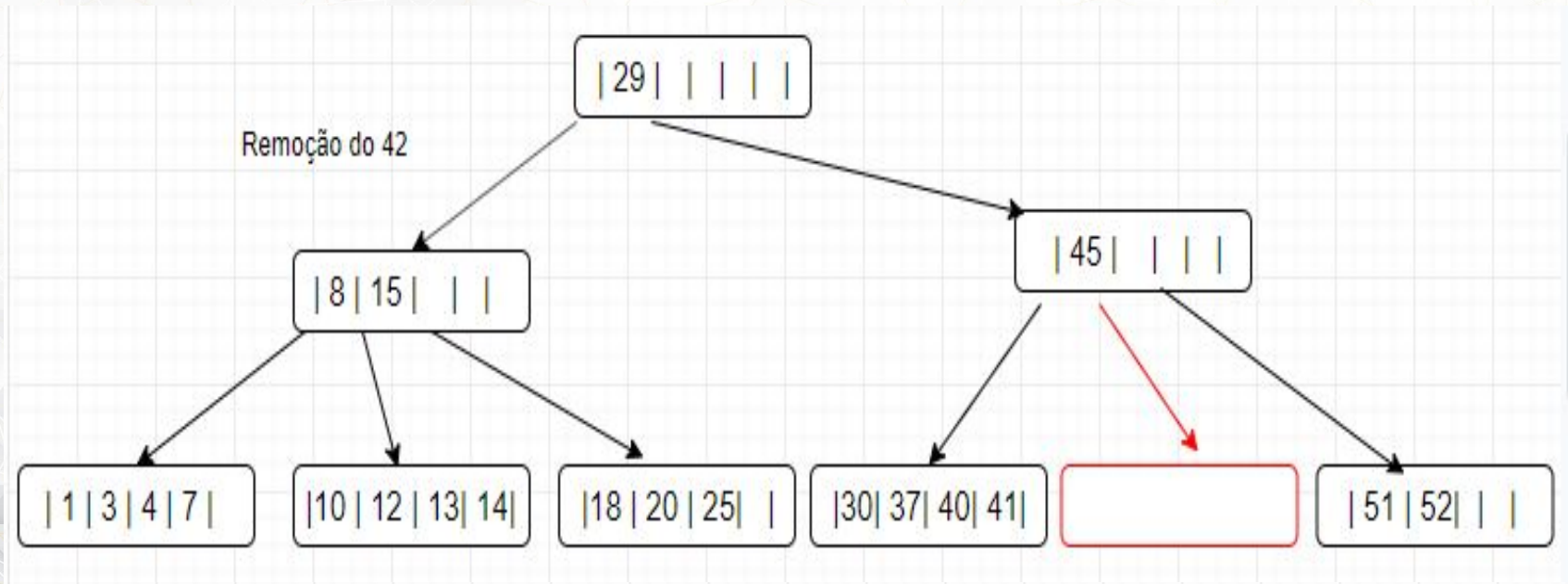


# Remoção em árvore B

- Caso 4: se a folha ficar com menos de 50% de ocupação, e as páginas irmãs não puderem ceder uma chave

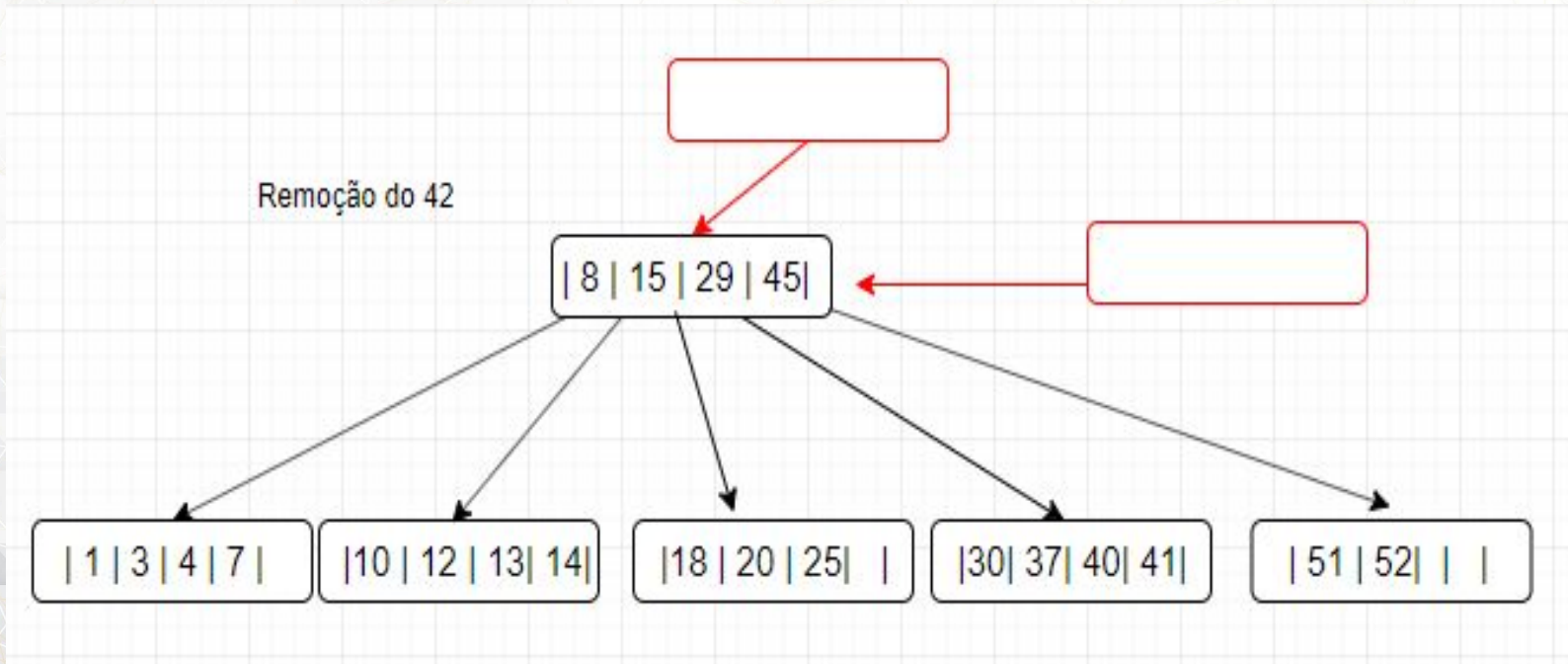


# Remoção em árvore B



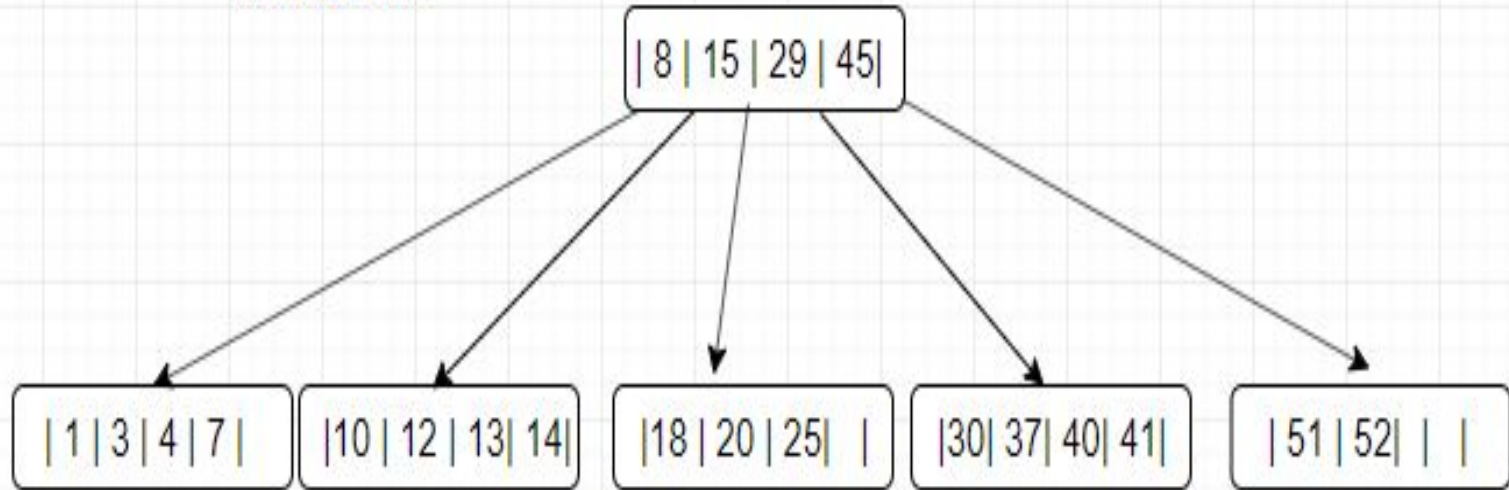


# Remoção em árvore B



# Remoção em árvore B

Remoção do 42



# Dúvidas?

Email:

[ricardo.de.oliveira96@gmail.com](mailto:ricardo.de.oliveira96@gmail.com)

[lucas\\_god1997@alunos.utfpr.edu.br](mailto:lucas_god1997@alunos.utfpr.edu.br)

[olivelton00@gmail.com](mailto:olivelton00@gmail.com)

