



# Algoritmo e Estrutura de Dados II

## COM-112

Aula 16

Vanessa Souza

# Operações Básicas sobre a árvore B



# Operações Básicas

---

- ▶ Busca
- ▶ Inserção
- ▶ Remoção





# Operações Básicas

---

- ▶ Busca
- ▶ Inserção
- ▶ Remoção



# Remoção em Árvore B



## Remoção em Árvore B

---

- ▶ A remoção em árvore B segue a mesma ideia da remoção em árvores binárias.
- ▶ No entanto, a remoção de um elemento de uma árvore B pode ser dividida em dois casos:
  - ▶ O elemento que será removido está em uma folha
  - ▶ O elemento que será removido está em um nó interno.
- ▶ Em qualquer um dos casos, uma coisa importante é saber a quantidade de chaves que restaram no nó após a remoção.





## Remoção em Árvore B

---

- ▶ Se a **folha** contém mais que o número mínimo de entradas então não há mais nenhuma ação.
  - ▶ Simplesmente elimine o elemento da folha
- ▶ Caso contrário será preciso emprestar do pai OU unir nós.





## Remoção em Árvore B

---

- ▶ Podem ocorrer duas situações :
- ▶ 1) Após a remoção, o nó ainda possui uma quantidade adequada de dados
- ▶ 2) Após a remoção, o nó ultrapassou o limite mínimo de dados. Neste caso, pode-se tentar alguma das opções abaixo:
  - a) Tente emprestar um nó do irmão IMEDIATAMENTE à esquerda;
  - b) Se não for possível, tente emprestar do irmão da DIREITA; ou
  - c) Se ambas falharem, fazer a fusão do nó com um de seus irmãos, incluindo o pai.
    - a) Verifica se o pai 'desbalanceou' e repete o processo para o pai







## Remoção em Árvore B

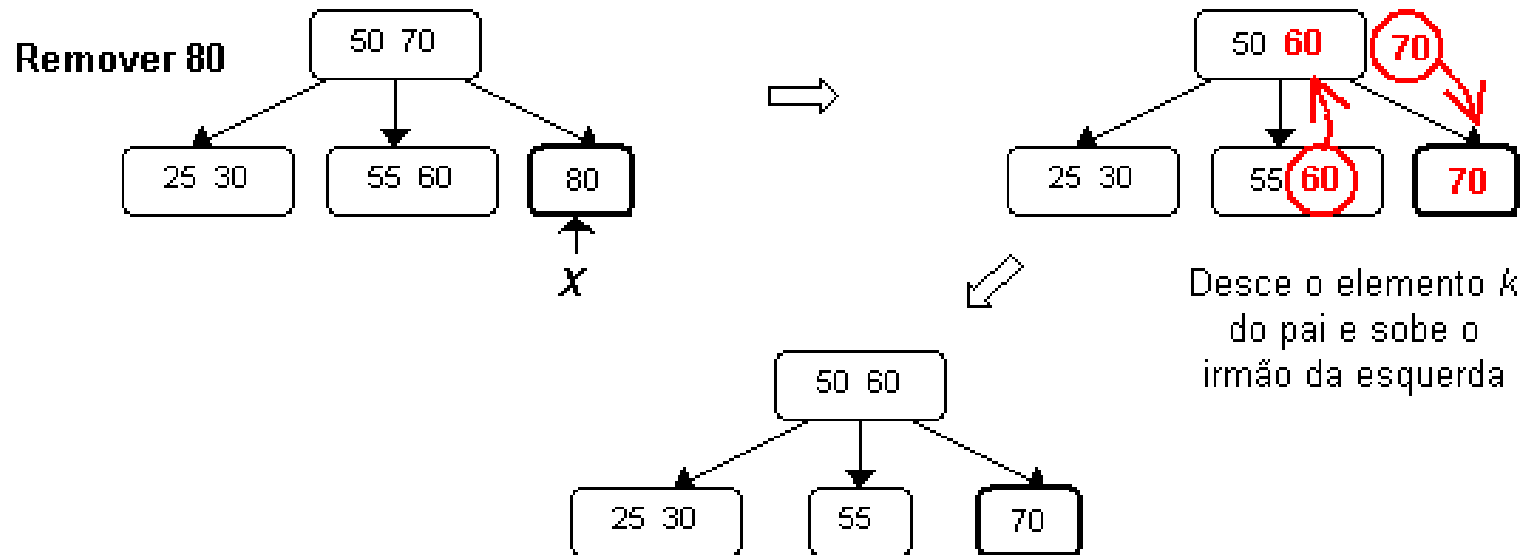
---

- ▶ Quando um elemento for removido de uma folha ***X*** e o número de elementos no nó folha diminui para menos que ***t - 1***, deve-se reorganizar a árvore B.
- ▶ A solução mais simples é analisarmos os irmãos da direita ou esquerda de ***X***.
  - ▶ Se um dos irmãos (da direita ou esquerda) de ***X*** possui mais de ***t - 1*** elementos, a chave ***k*** do pai que separa os irmãos pode ser incluída no nó ***X*** e a última ou primeira chave do irmão (última se o irmão for da esquerda e primeira se o irmão for da direita) pode ser inserida no pai no lugar de ***k***.





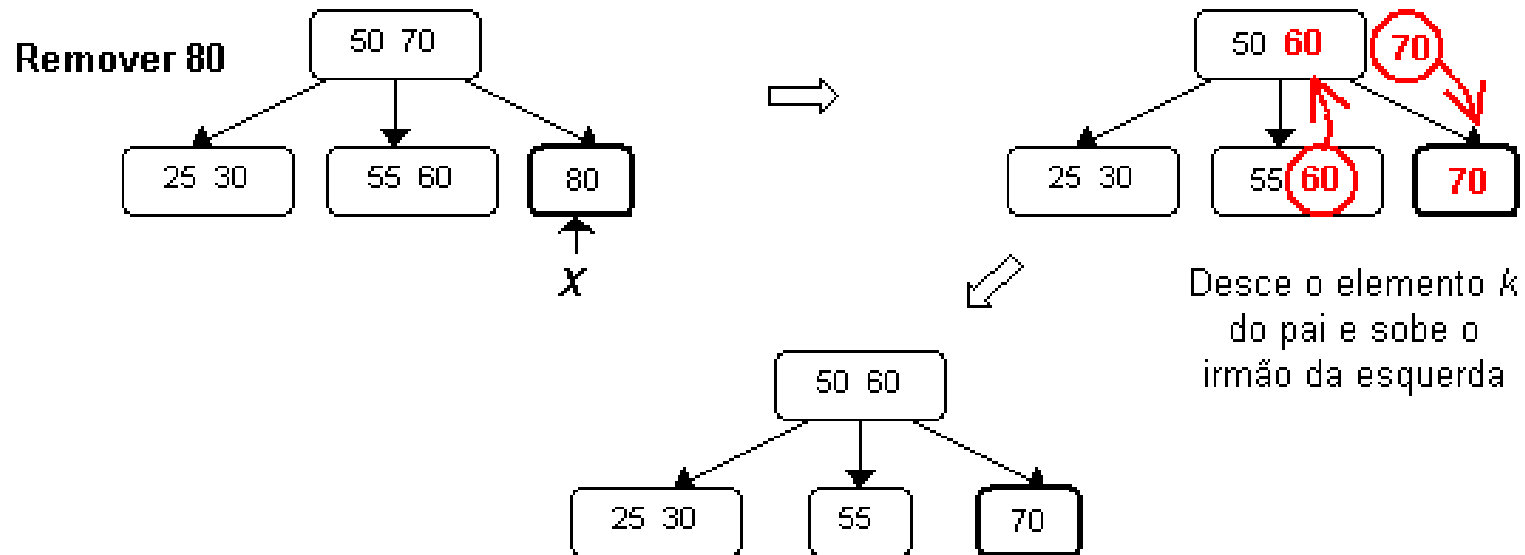
## Remoção em Árvore B



Passos para a remoção da chave 80 em uma árvore B com  $t = 2$



## Remoção em Árvore B



Passos para a remoção da chave 80 em uma árvore B com  $t = 2$

ESSA É A OPERAÇÃO DE ROTAÇÃO EM ÁRVORES B.



## Remoção em Árvore B

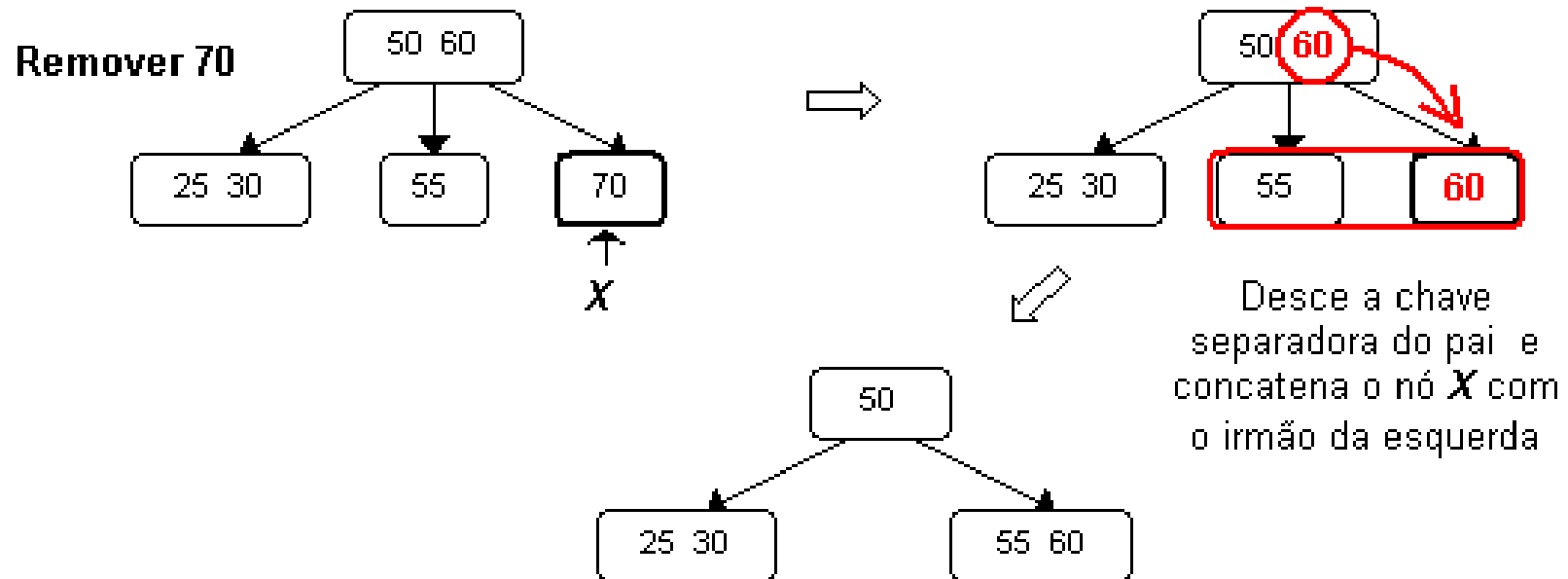
---

- ▶ Quando um elemento for removido de uma folha ***X*** e o número de elementos no nó folha diminui para menos que ***t - 1***, deve-se reorganizar a árvore B.
- ▶ A solução mais simples é analisarmos os irmãos da direita ou esquerda de ***X***.
  - ▶ Se os dois irmãos de ***X*** contiverem exatamente ***t - 1*** elementos (ocupação mínima), nenhum elemento poderá ser emprestado.
  - ▶ Neste caso, o nó ***X*** e um de seus irmãos são concatenados em um único nó que também contém a chave separadora do pai.





## Remoção em Árvore B



Passos para a remoção da chave 70 em uma árvore B com  $t = 2$

**ESSA É A OPERAÇÃO DE MERGE EM ÁRVORES B.**



## Remoção em Árvore B

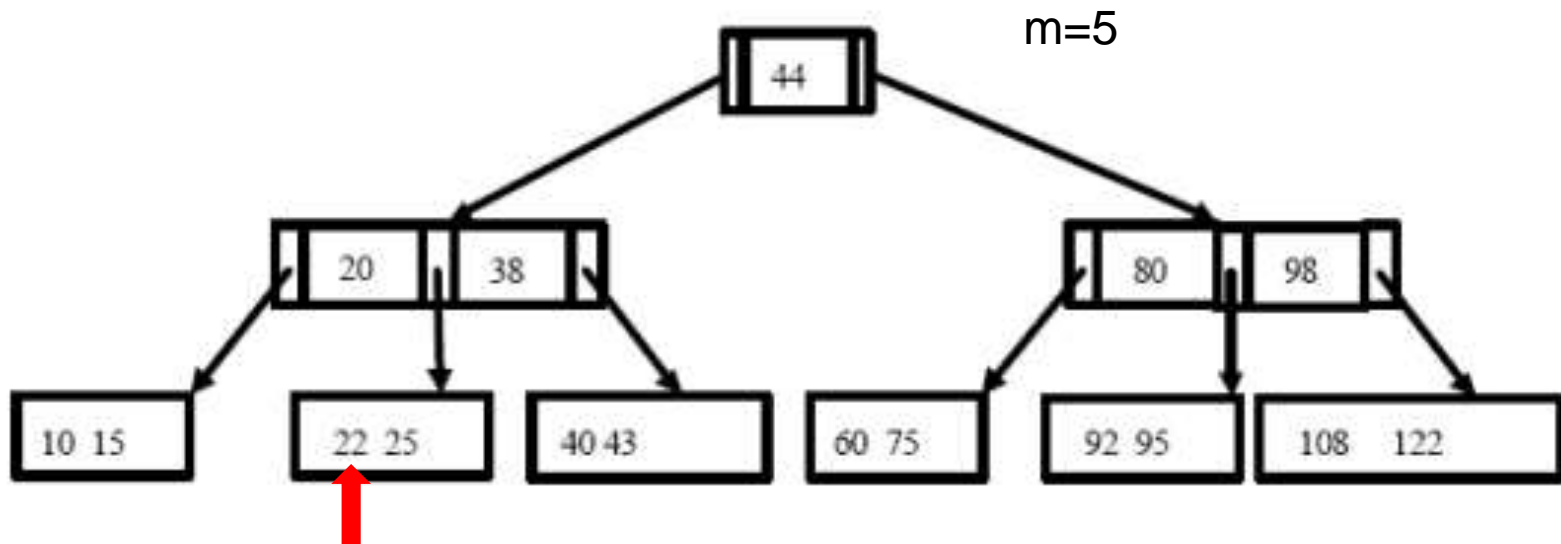
---

- ▶ Quando um elemento for removido de uma folha ***X*** e o número de elementos no nó folha diminui para menos que ***t - 1***, deve-se reorganizar a árvore B.
- ▶ A solução mais simples é analisarmos os irmãos da direita ou esquerda de ***X***.
  - ▶ Se o pai também contiver apenas ***t - 1*** elementos, deve-se considerar os irmãos do pai como no caso anterior e proceder recursivamente.





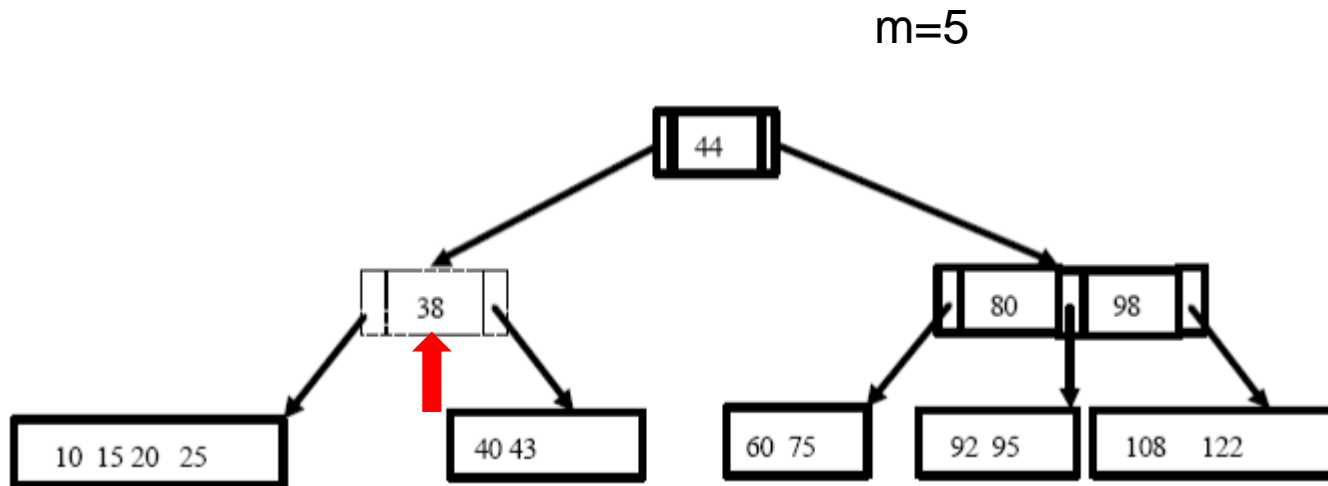
# Remoção em Árvore B



Remover nó 22



# Remoção em Árvore B



Remover nó 22



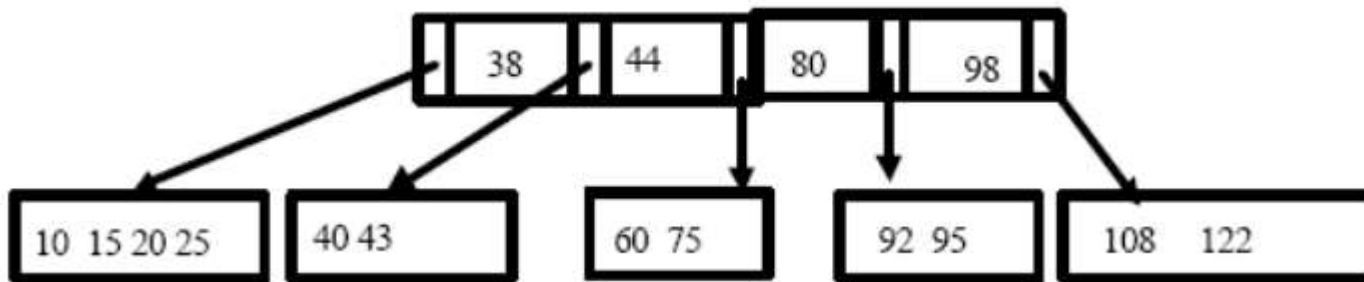




## Remoção em Árvore B

---

$m=5$



Árvore final após a remoção do nó 22



## Remoção em Árvore B

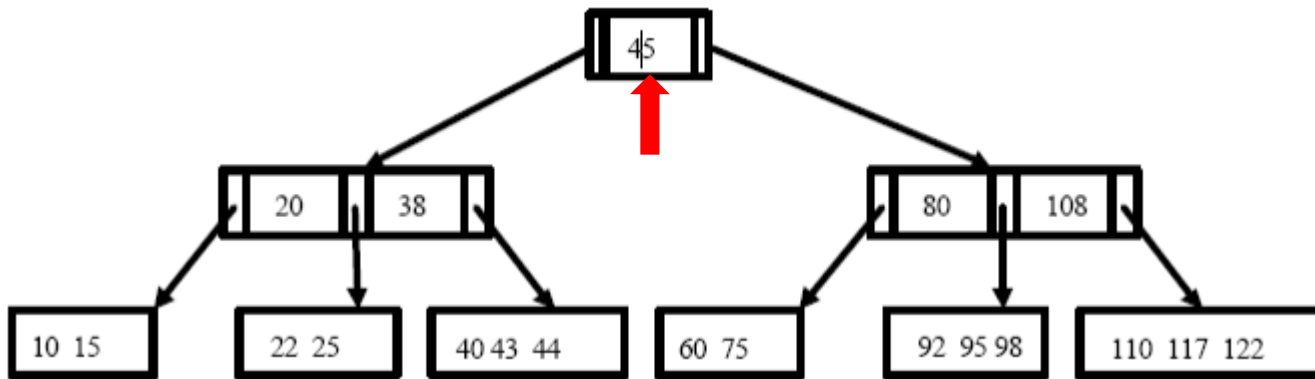
---

- ▶ Se o elemento estiver sendo removido de um nó não folha:
  - ▶ Movemos o sucessor/predecessor do elemento, que deve estar em uma folha, para a posição eliminada.
  - ▶ O caso passa a ser remover da folha.



# Remoção em Árvore B

$m=5$



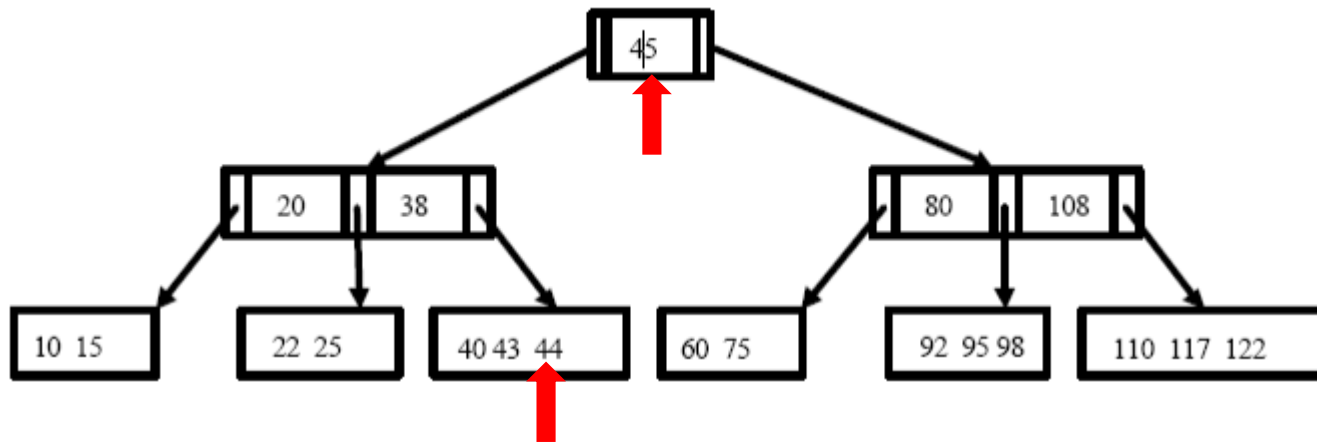
Remover nó 45





# Remoção em Árvore B

$m=5$

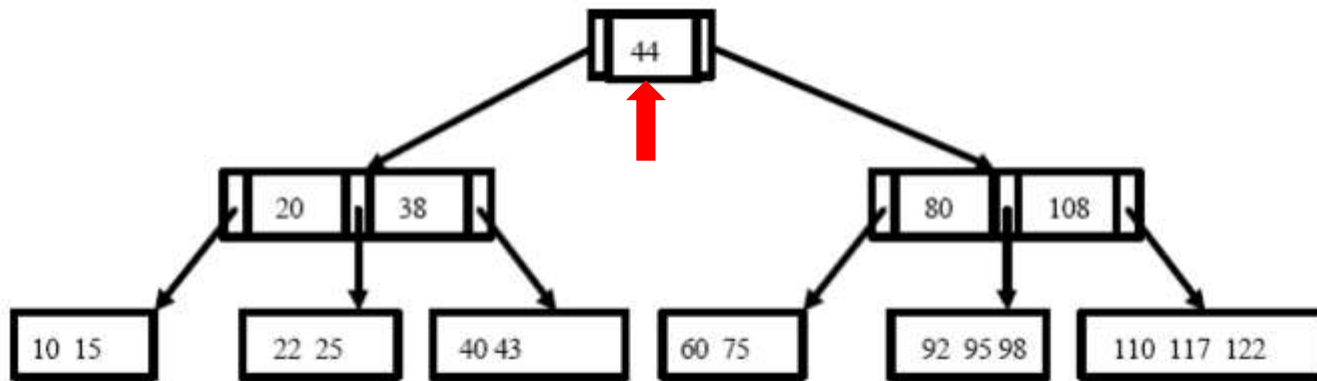


Remover nó 45



# Remoção em Árvore B

$m=5$



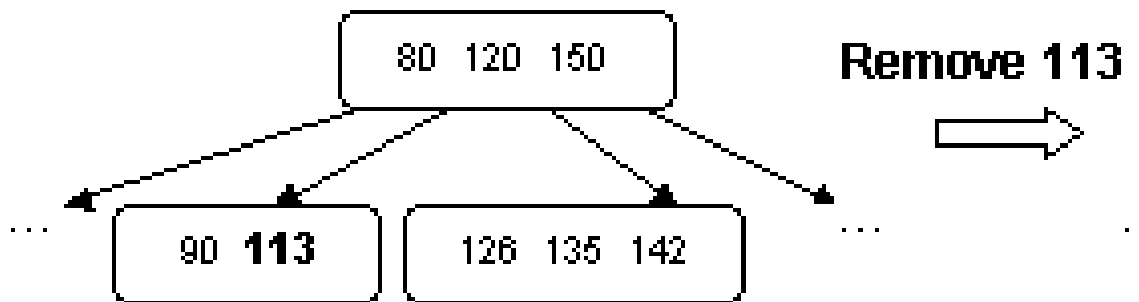
Remover nó 45





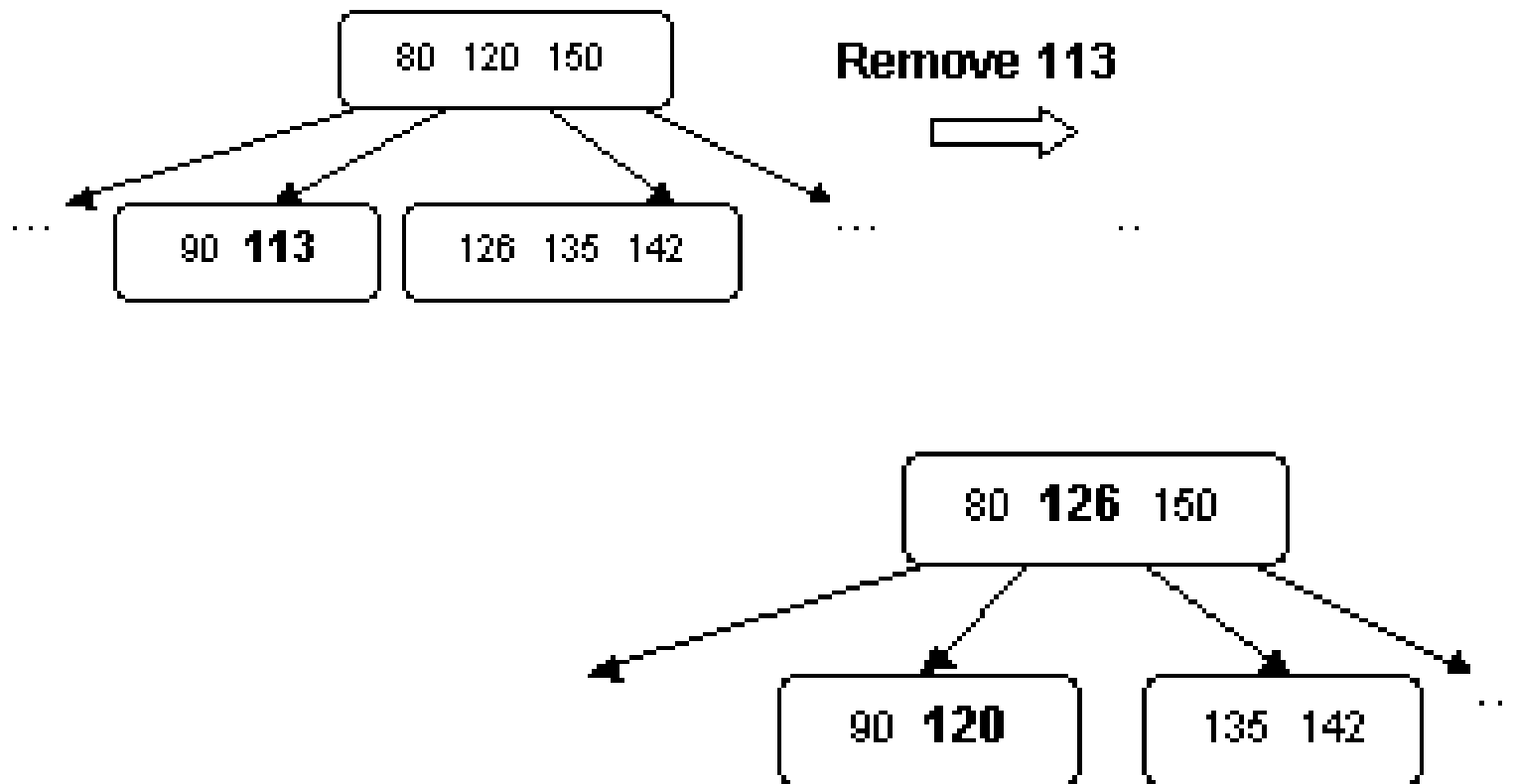
## Exemplo 1 : Remoção em Árvore B – $t = 3$

---



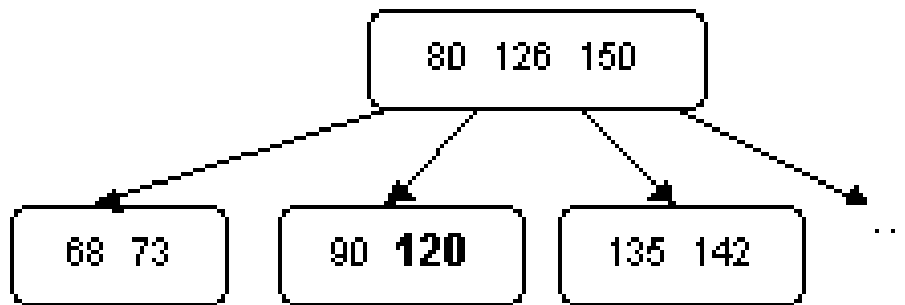


## Exemplo 1 : Remoção em Árvore B – $t = 3$

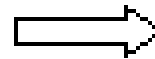




## Exemplo 2 : Remoção em Árvore B – $t = 3$



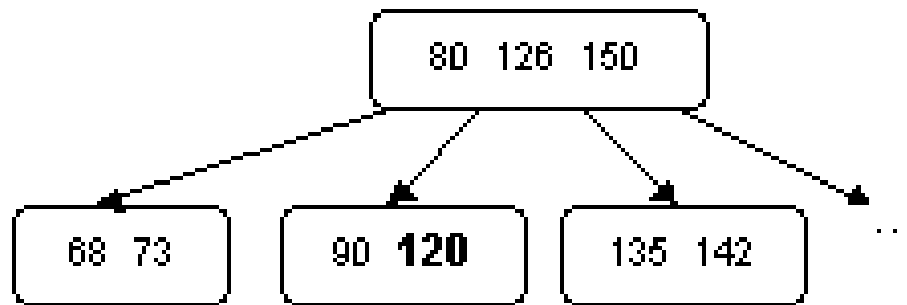
Remove 120



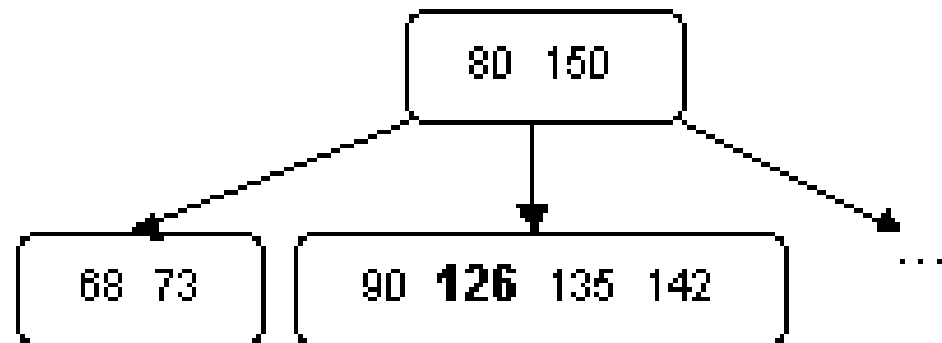
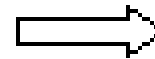




## Exemplo 2 : Remoção em Árvore B – $t = 3$

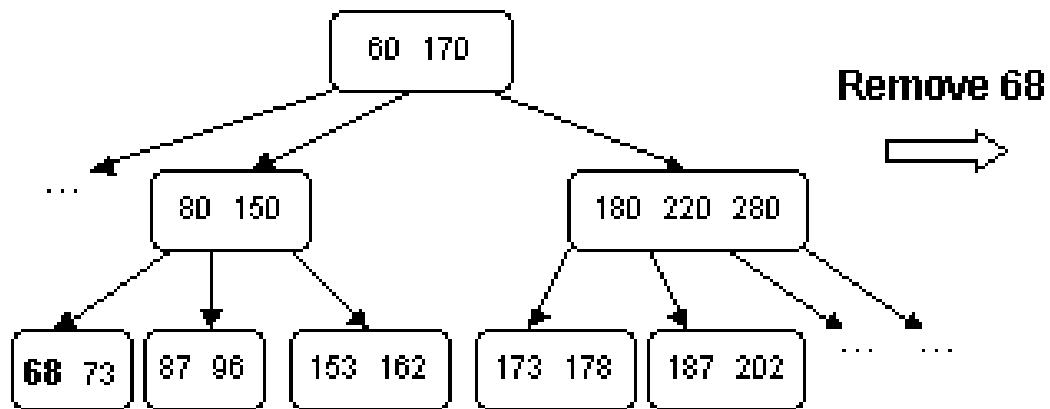


Remove 120



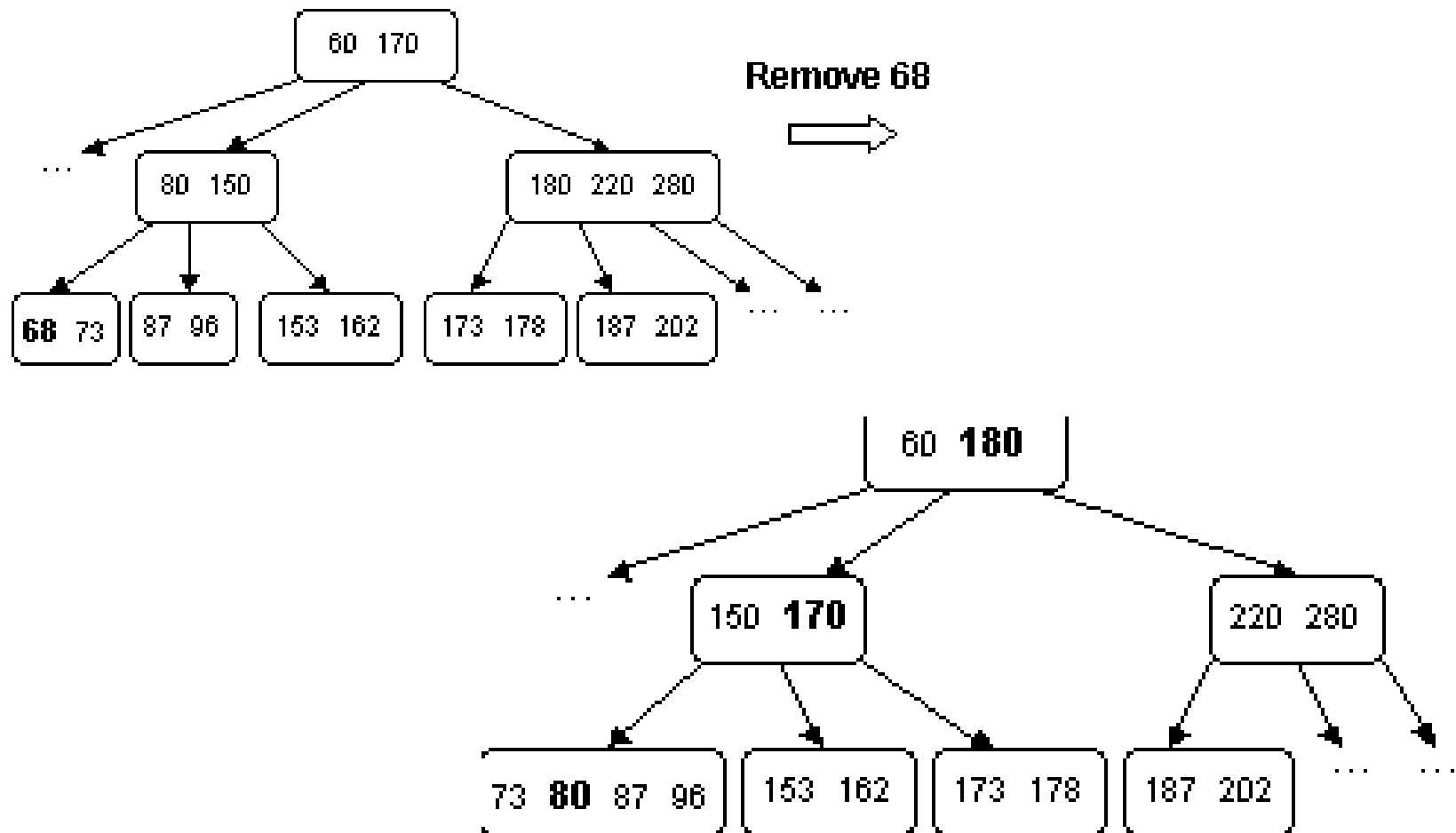


## Exemplo 3 : Remoção em Árvore B – $t = 3$



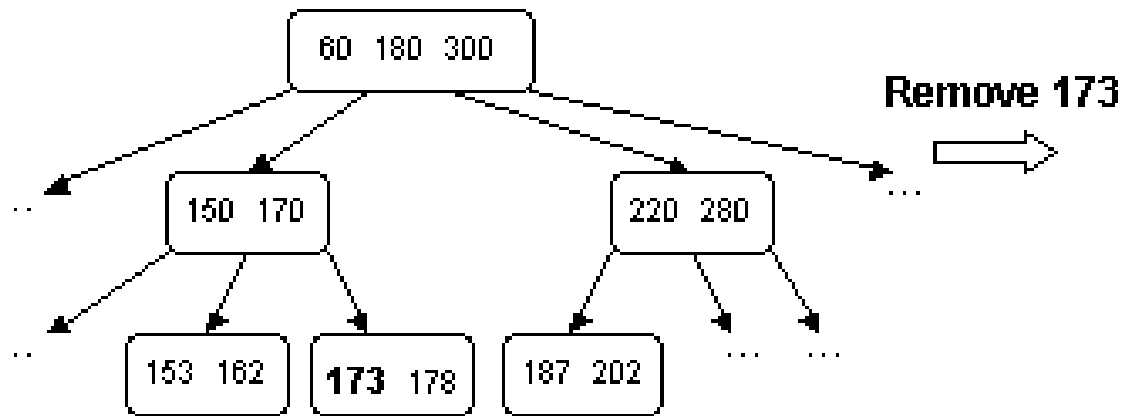


## Exemplo 3 : Remoção em Árvore B – t = 3



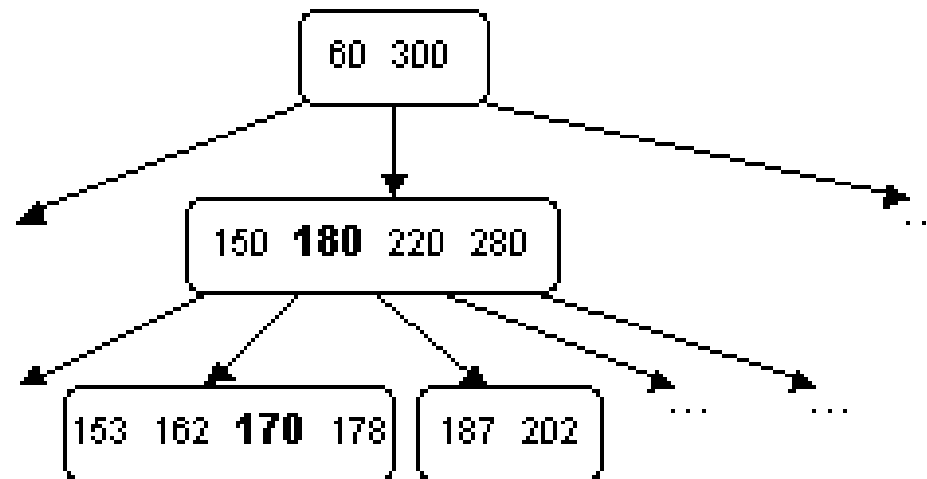
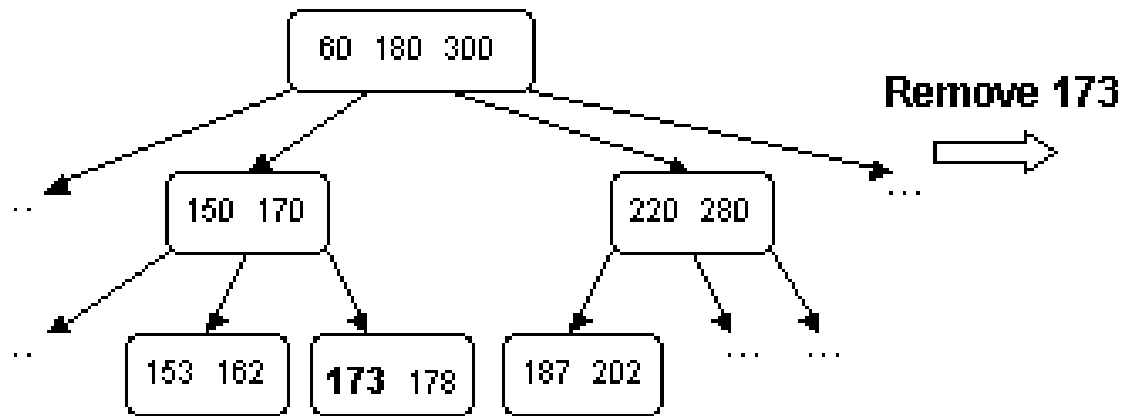


## Exemplo 4 : Remoção em Árvore B – $t = 3$





## Exemplo 4 : Remoção em Árvore B – t = 3

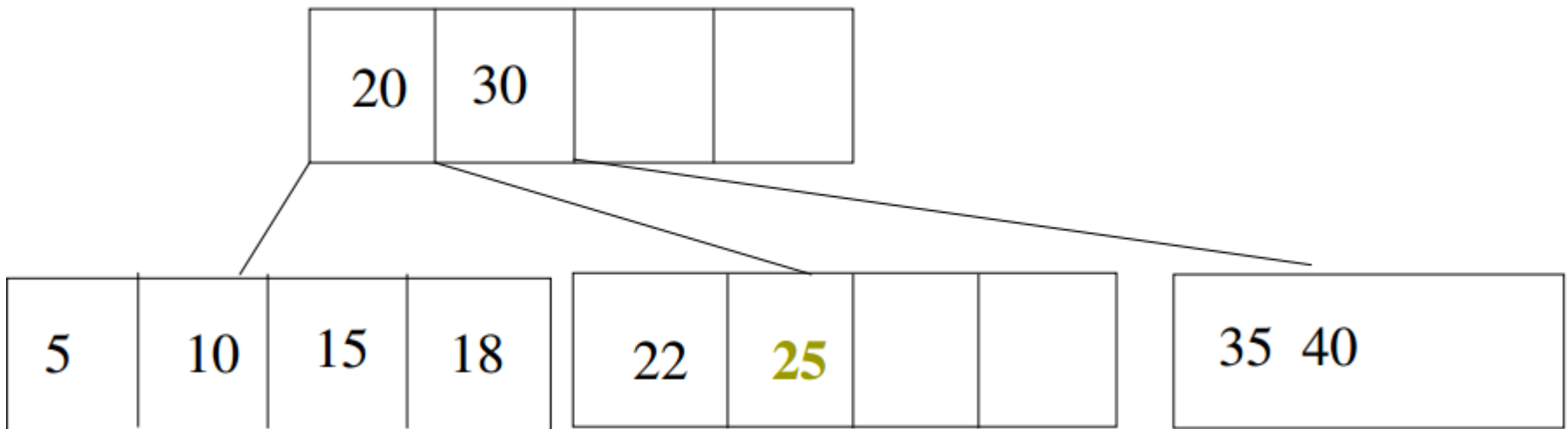




## Exemplo 5 : Remoção em Árvore B – ordem 5

---

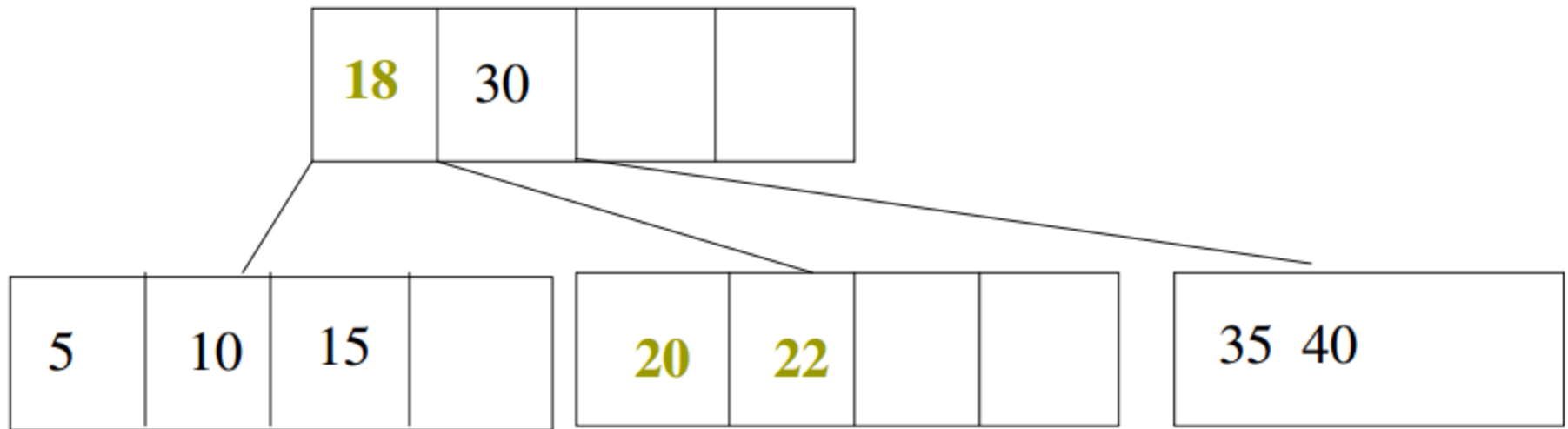
Remover 25





## Exemplo 5 : Remoção em Árvore B – ordem 5

---





## Remoção – algoritmo em alto nível

---

1. Localizar a chave
2. Se a chave está em uma folha, remova chave
3. Se chave não está em uma folha, substitua por maior chave da subárvore esquerda (ou menor da subárvore direita).
4. Se nó não fica com menos de  $m/2-1$  chaves – terminou
5. Senão:
  1. Redistribuir chaves entre os nós vizinhos OU concatenar nó com um nó vizinho e chave no pai entre os 2 nós filhos e verificar o pai.



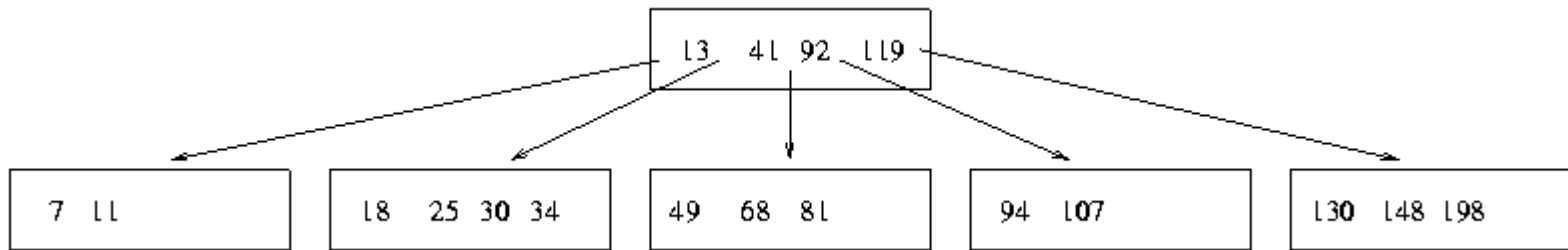




## Exercícios

---

- ▶ Dada a árvore B (com  $m=5$ ) mostrada abaixo, desenhe as árvores resultantes da aplicação das seguintes operações: inserção de 98, inserção de 28, remoção de 94 e remoção de 13.

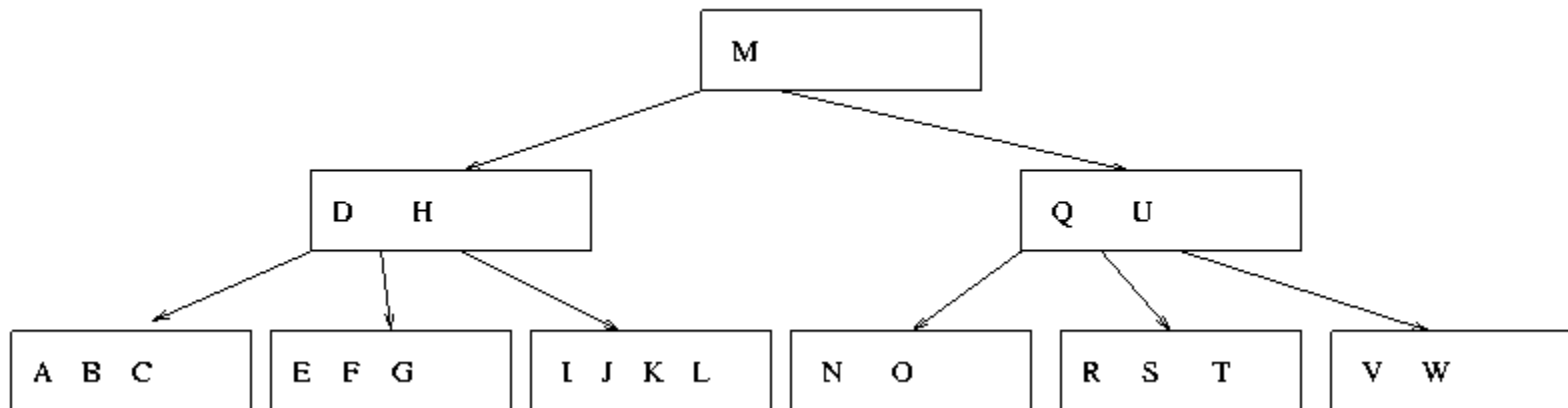




## Exercícios

---

- ▶ Mostrar as árvores B resultantes da remoção de A, B, Q e M, nesta ordem.



Árvore B de Ordem 6