



# Árvores-B (Parte IIIa)

---

SCC-203 – Algoritmos e Estruturas de Dados II

Graça Nunes



# Propriedades das árvores-B

---

- Relembrando...

- Para uma árvore-B de ordem **m**

1. cada página tem, no máximo, **m** descendentes
2. cada página, exceto a raiz e as folhas, tem no mínimo  $\lceil \mathbf{m/2} \rceil$  descendentes
3. a raiz tem, no mínimo, dois descendentes - a menos que seja uma folha
4. todas as folhas estão no mesmo nível
5. uma página não folha que possui **k** descendentes contém **k-1** chaves
6. uma página folha contém, no mínimo  $\lceil \mathbf{m/2} \rceil - \mathbf{1}$  e, no máximo, **m-1** chaves



# Eliminação, Redistribuição e Concatenação

---

- O *split* garante a *manutenção das propriedades da árvore-B* durante a inserção
- Essas **propriedades precisam ser mantidas**, também, **durante a eliminação** de chaves
- Há vários casos para se analisar (árvore de ordem **m**)



# Eliminação: Caso 1

---

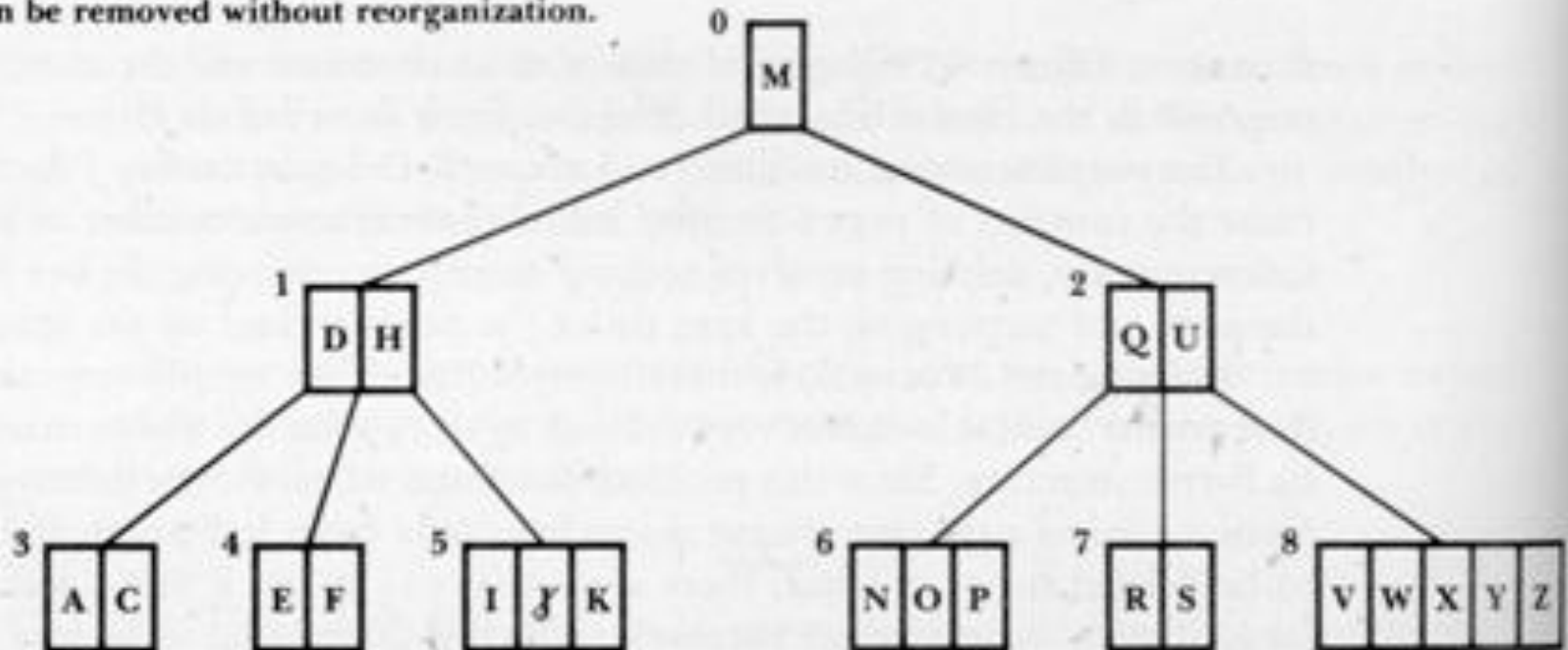
- **Caso 1:** eliminação de uma chave em uma página folha, sendo que o número mínimo de chaves na página é respeitado:  $\lceil m/2 \rceil - 1$
- **Solução:** chave é retirada e os registros internos à página são reorganizados

# Eliminação: Caso 1

$m = 6$   
 $m-1=5$  (max)  
 $m/2 = 3$  (min)  
 $m/2 - 1 = 2$  (min)

## Case 1: No action.

Delete  $J$  from page 5. Since page 5 has more than the minimum number of keys,  $J$  can be removed without reorganization.





# Eliminação: Caso 2

---

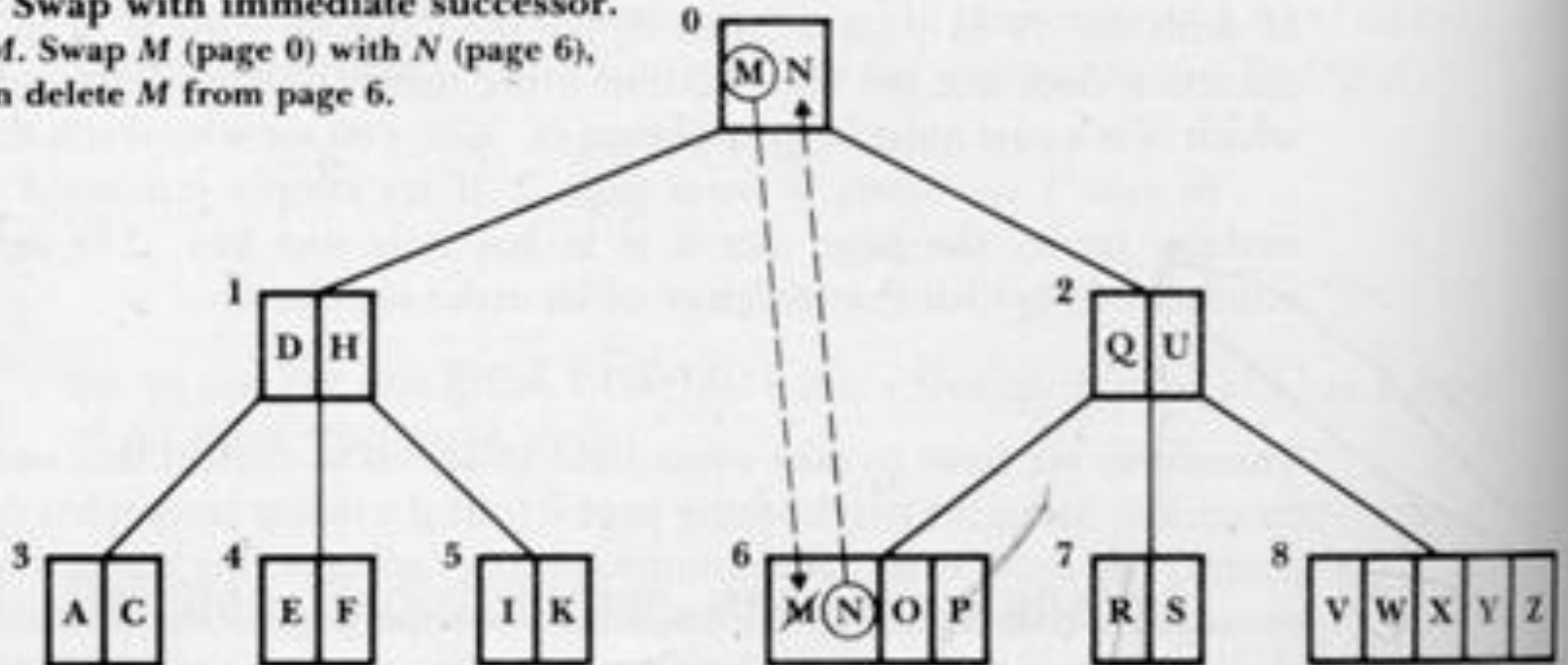
- **Caso 2:** eliminação de uma **chave que não está em uma folha**
- **Solução:** **sempre eliminamos de páginas folha\***
  - Se uma chave deve ser eliminada de uma página que não é folha, trocamos a chave com sua sucessora imediata (ou com a predecessora imediata) que está numa folha
  - A seguir, eliminamos a chave da folha

\* Análogo a AVL

# Eliminação: Caso 2

$m = 6$   
 $m-1=5$  (max)  
 $m/2 = 3$  (min)  
 $m/2 - 1 = 2$  (min)

**Case 2: Swap with immediate successor.**  
Delete  $M$ . Swap  $M$  (page 0) with  $N$  (page 6),  
and then delete  $M$  from page 6.





# Eliminação: Caso 3

---

- **Caso 3:** eliminação causa *underflow* na página (folha)
- **Solução:** redistribuição
  - Procura-se uma página irmã (mesmo pai) que contenha mais chaves do que o mínimo: se existir, redistribuem-se as chaves entre essas páginas
  - A redistribuição pode provocar uma alteração na chave separadora que está no nó pai

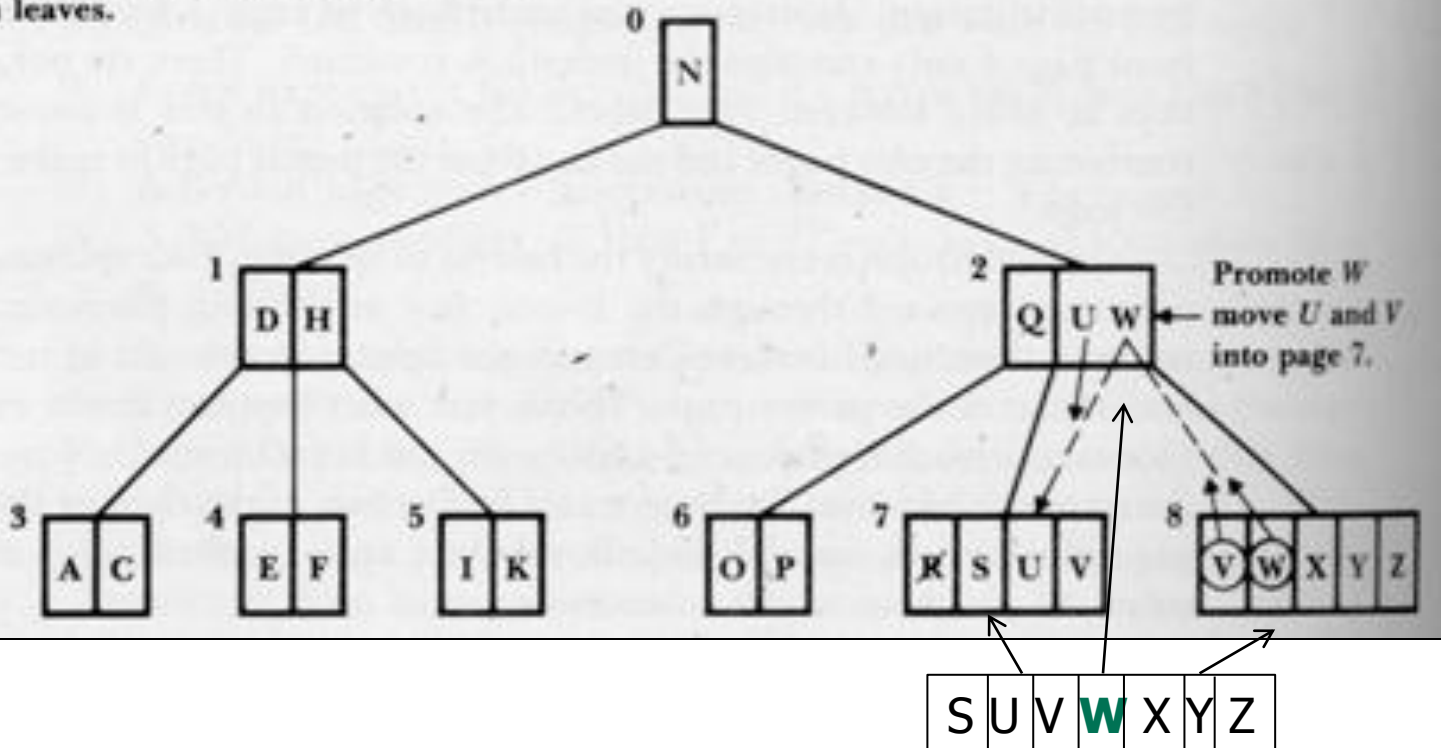


# Eliminação: Caso 3

$m = 6$   
 $m-1=5$  (max)  
 $m/2 = 3$  (min)  
 $m/2 - 1 = 2$  (min)

## Case 3: Redistribution.

Delete *R*. Underflow occurs. Redistribute keys among pages 2, 7, and 8 to restore balance between leaves.





# Eliminação: Caso 4

---

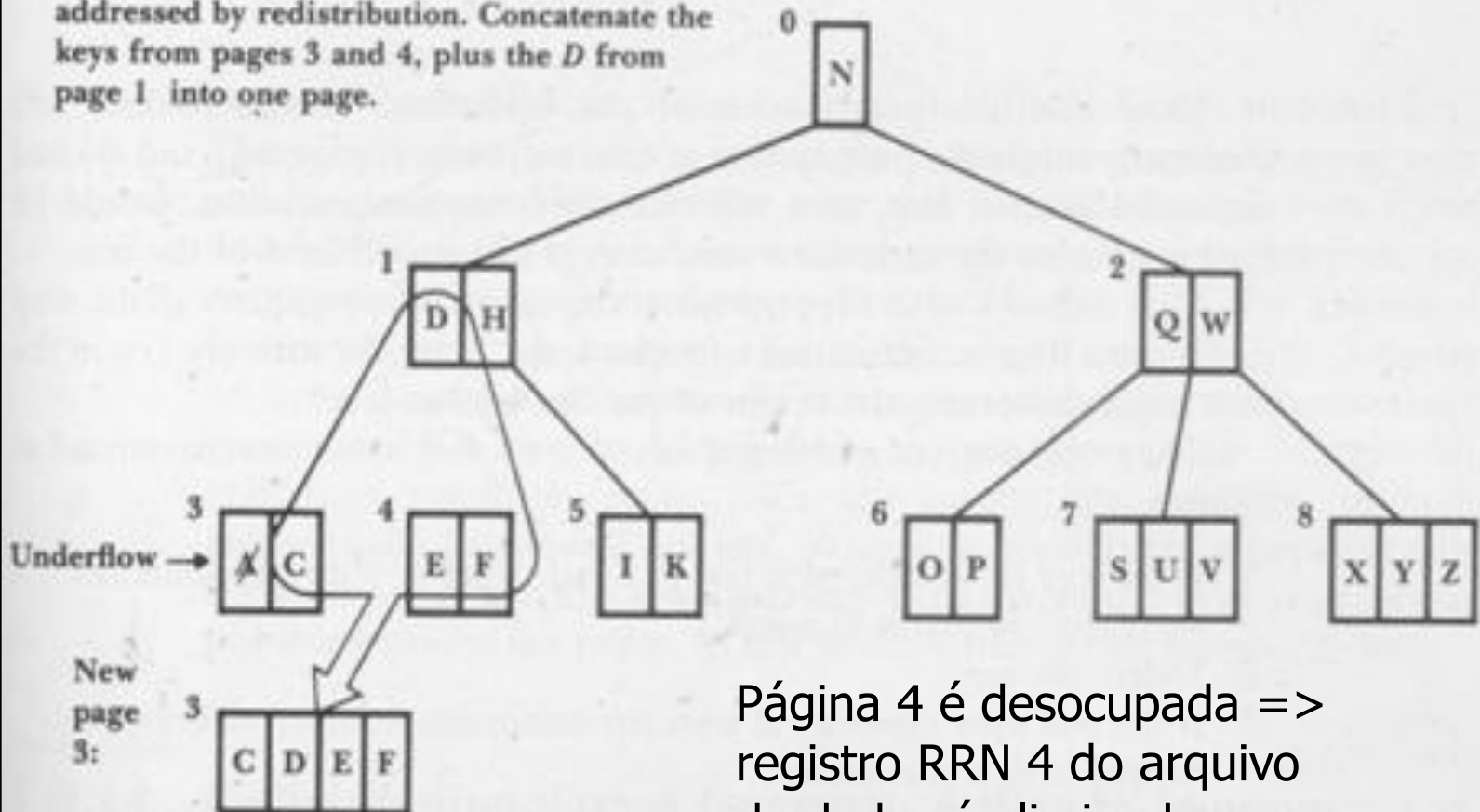
- **Caso 4:** ocorre *underflow* e a redistribuição não pode ser aplicada
  - Não existem chaves suficientes para dividir entre as duas páginas irmãs. A que sofreu *underflow* tem  $m/2-2$  chaves, e a outra,  $m/2-1$  chaves
- **Solução:** concatenação
  - Combina-se o conteúdo das duas páginas ( $m-3$ ) mais a chave separadora da página pai para formar uma única página com  $m-2$  chaves
  - A concatenação é o inverso do processo de particionamento
    - Como consequência, a eliminação na página pai tb pode causar *underflow*
    - Uma página é liberada (registro eliminado do arquivo de dados)

# Eliminação: Caso 4

$m = 6$   
 $m-1=5$  (max)  
 $m/2 = 3$  (min)  
 $m/2 - 1 = 2$  (min)

## Case 4: Concatenation.

Delete A. Underflow occurs, but it cannot be addressed by redistribution. Concatenate the keys from pages 3 and 4, plus the D from page 1 into one page.



Página 4 é desocupada =>  
registro RRN 4 do arquivo  
de dados é eliminado



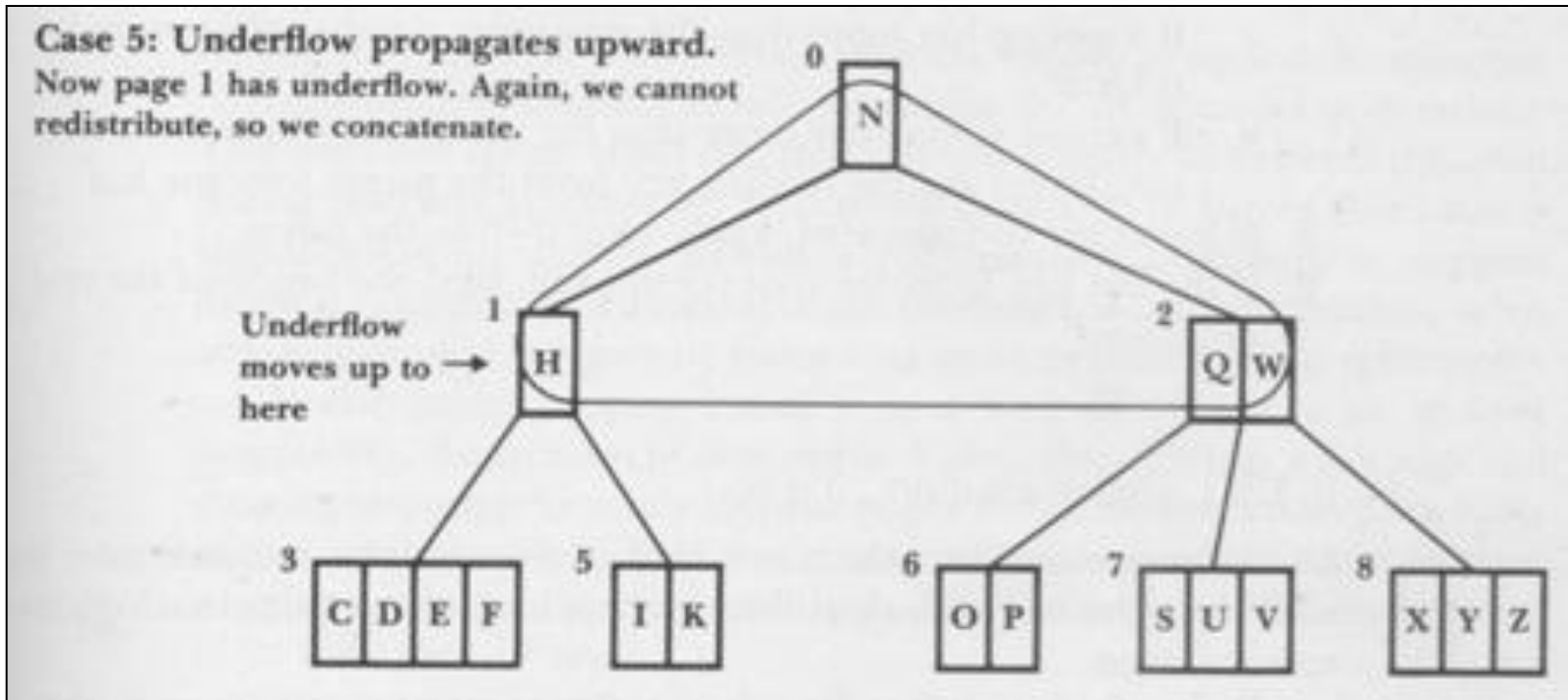
# Eliminação: Caso 5

---

- **Caso 5:** *underflow* da página pai, como consequência da concatenação
- **Solução:** utiliza-se redistribuição ou concatenação novamente

# Eliminação: Caso 5

$m = 6$   
 $m-1=5$  (max)  
 $m/2 = 3$  (min)  
 $m/2 - 1 = 2$  (min)





# Eliminação: Caso 6

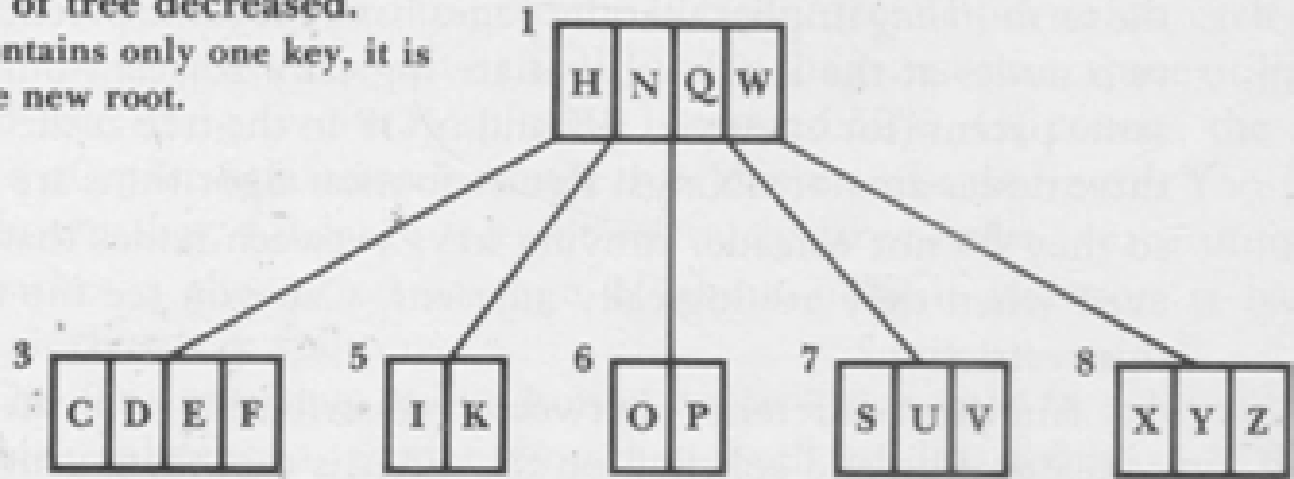
---

- **Caso 6:** **diminuição da altura da árvore**
  - Ocorre quando o nó raiz tem uma única chave
- **Solução:** **concatenação** nos seus nós filhos

# Eliminação: Caso 6

$m = 6$   
 $m-1=5$  (max)  
 $m/2 = 3$  (min)  
 $m/2 -1 = 2$  (min)

Case 6: Height of tree decreased.  
Since the root contains only one key, it is absorbed into the new root.



Registros RRN 0 e 2 são eliminados do arquivo de dados



# Algoritmo para eliminação de chaves em árvores-B

---

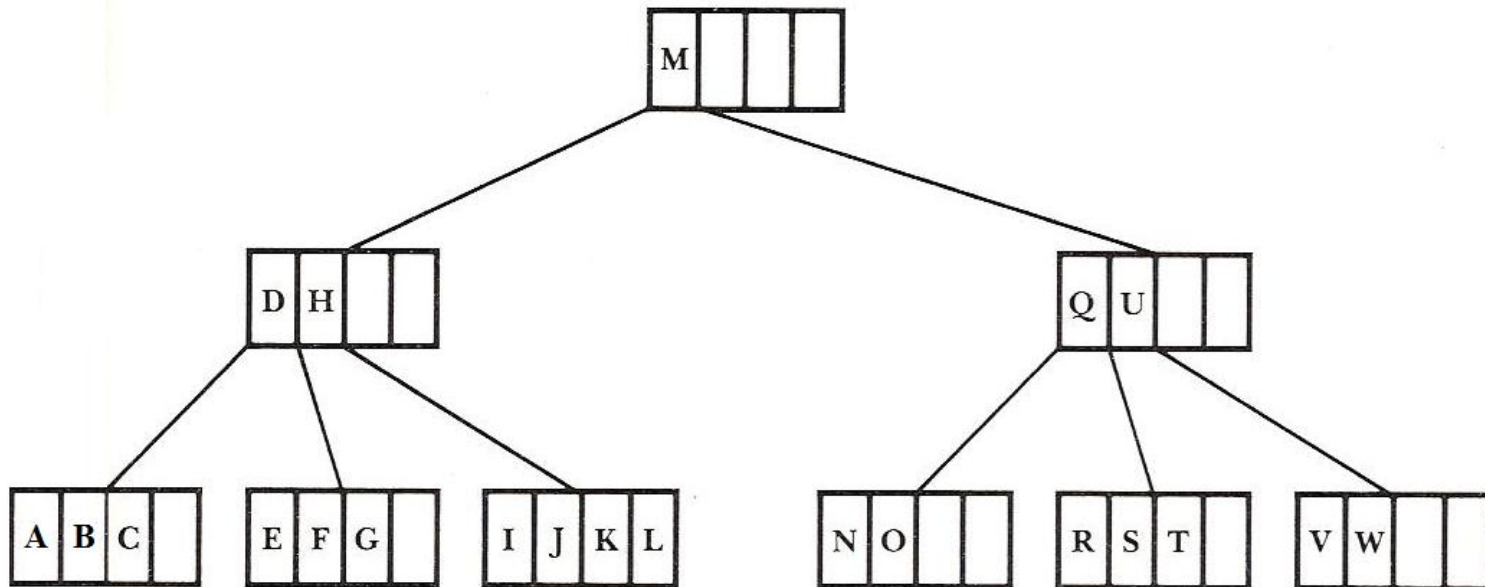
1. Se a chave não estiver numa folha, troque-a com sua sucessora\*
2. Elimine a chave da folha
3. Se a página continuar com o número mínimo de chaves, fim
4. Se a folha tem uma chave a menos que o mínimo, verifique as páginas irmãs a esquerda e a direita
  - 4.1. se uma delas tiver mais do que o número mínimo de chaves, aplique redistribuição
  - 4.2. senão concatene a página com uma das irmãs e a chave separadora do pai
5. Se ocorreu concatenação, aplique os passos de 3 a 6 para a página pai
6. Se a última chave da raiz for removida, a altura da árvore diminui

\*primeira chave da página mais à esquerda da filha à direita; ou última da página mais à direita da filha à esquerda



# Exercício

- Usando o algoritmo anterior, remova as chaves A, B, Q e R da árvore-B de ordem 5 abaixo





## 2º. Exercício para entregar

---

- Estenda o programa do 1º. Exercício com a implementação da função de eliminação de chave numa Árvore-B de ordem 8.
- Dada uma chave, a função deve retornar: Sucesso (eliminou) + RRN da página onde estava a chave; Fracasso (não eliminou)
- Teste sua função com a Árvore-B criada no 1º. Exercício.



# Exercício para entregar

---

- Função de Eliminação de chave em uma Árvore-B de ordem 8
- Dadas raiz da árvore e chave, retornar: Sucesso, RRN da página onde estava, e altura atual da árvore; ou Fracasso (chave não estava na árvore)
- Entregar por email (monitor e profa.) até o dia 18/6