

# Árvores-B: Remoção

Cristina Dutra de Aguiar Ciferri

Thiago A. S. Pardo

---

# Desempenho da Árvore-B

- Baseado em suas propriedades
  - 2 cada página, exceto a raiz e as folhas, possui no mínimo  $\lceil m/2 \rceil$  descendentes → taxa de ocupação
  - 5 uma página interna com  $k$  descendentes contém  $k-1$  chaves
  - 6 uma folha possui no mínimo  $\lceil m/2 \rceil - 1$  chaves e no máximo  $m - 1$  chaves → taxa de ocupação

habilidade de garantir que a árvore seja “larga e rasa” ao invés de “estreita e profunda”

---

# Desempenho da Árvore-B

- ***Split***
    - garante a manutenção das propriedades da árvore-B durante a inserção de novas chaves
  - Remoção
    - também deve garantir as propriedades durante a remoção de chaves
    - *underflow*
      - ocorre quando o número de chaves em uma página fica abaixo do número mínimo de chaves permitido pela árvore-B
-

# Situações (Casos)

1. Remoção de uma chave em um nó folha, sem causar *underflow*
  2. Remoção de uma chave em um nó não folha
  3. Remoção de uma chave em um nó, causando *underflow*
  4. Remoção de uma chave em um nó, causando *underflow* e a redistribuição não pode ser aplicada
  5. *Underflow* no nó pai causado pela remoção de uma chave em um nó filho
  6. Diminuição da altura da árvore
-

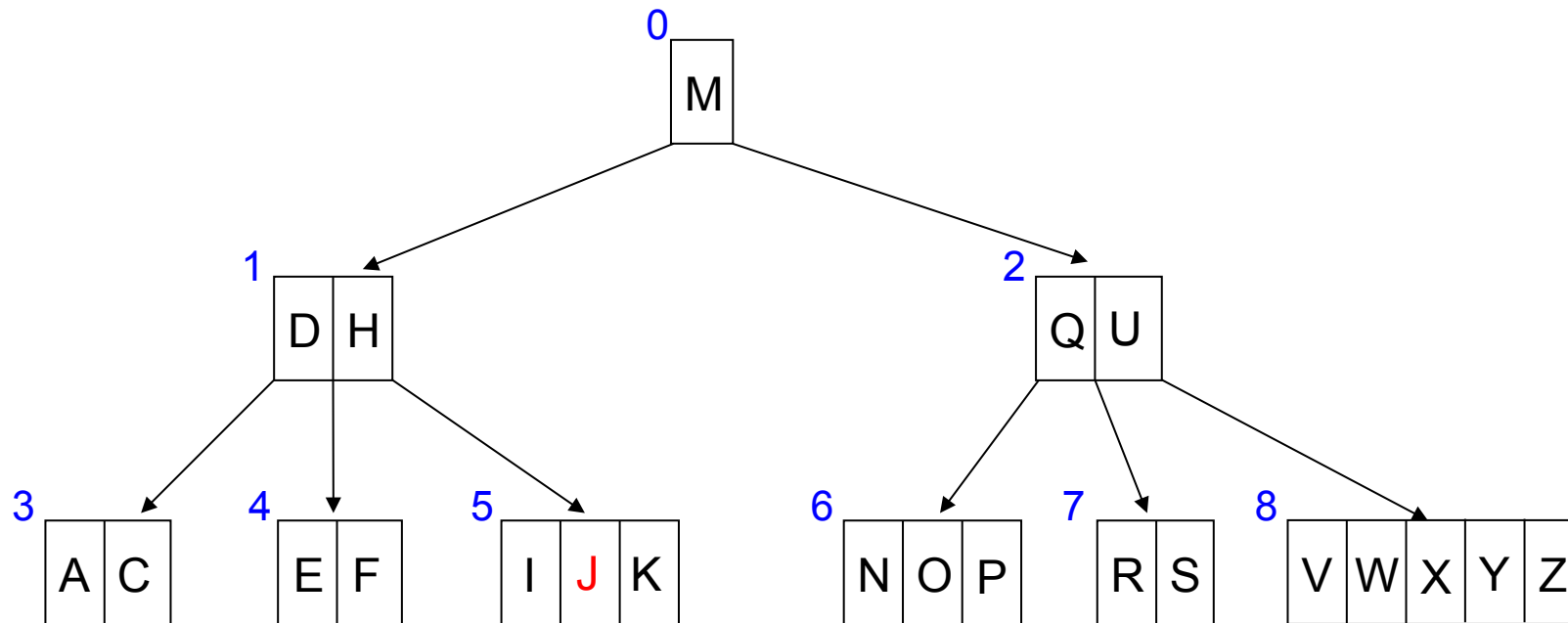
# Remoção: Caso 1

- Remoção de uma chave em um nó folha, sem causar *underflow*
    - situação mais simples possível
  - Solução
    - eliminar a chave da página
    - rearranjar as chaves remanescentes dentro da página para fechar o espaço liberado
-

# Remoção: Caso 1 (1/2)

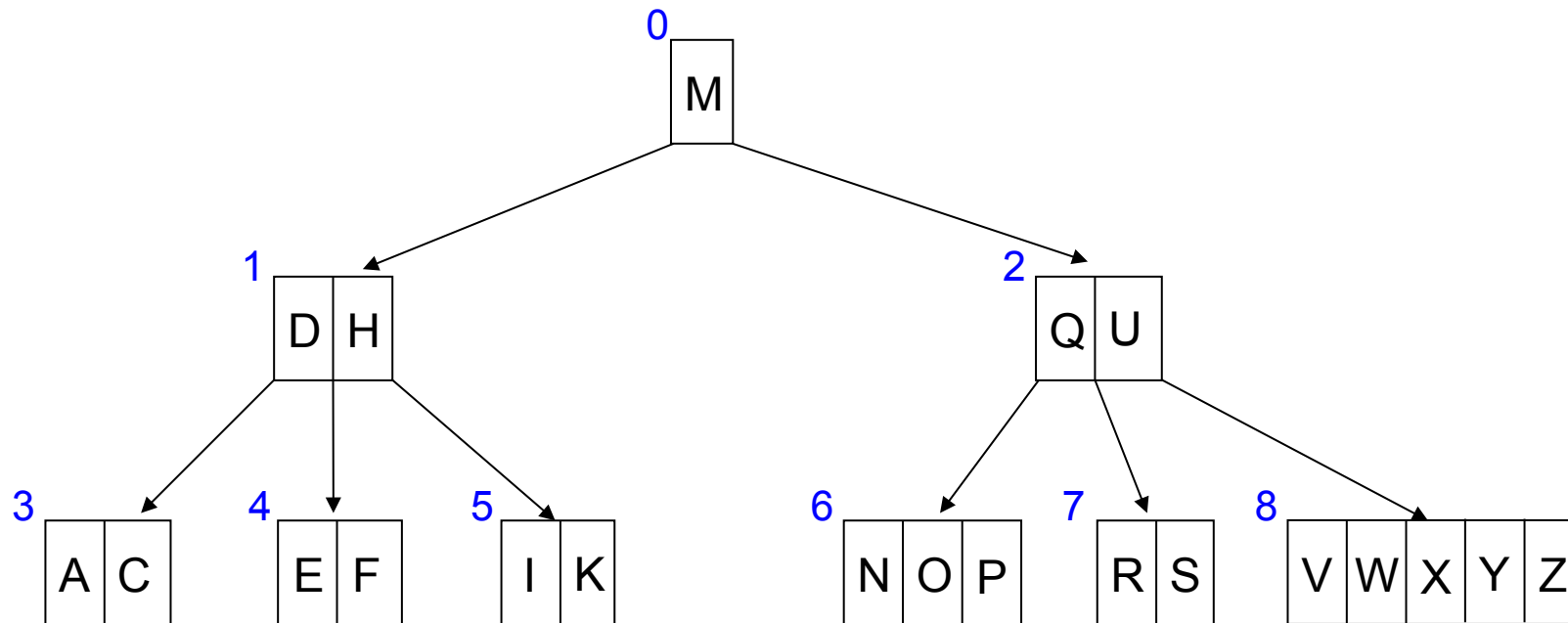
- Remoção de J

• árvore-B de ordem 6



# Remoção: Caso 1 (2/2)

- Remoção de J
  - página 5 garante a taxa de ocupação



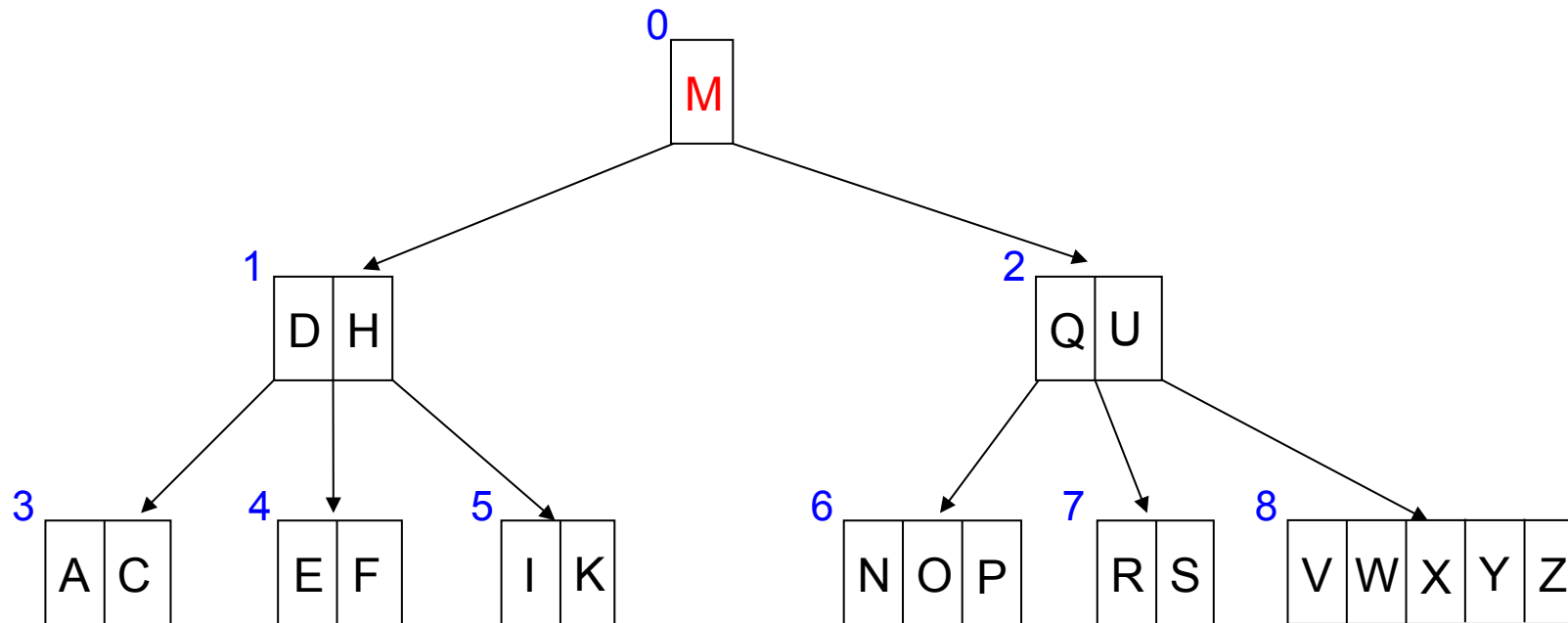
# Remoção: Caso 2

- Remoção de uma chave em um nó não folha
  - Solução
    - sempre remover chaves somente nas folhas
  - Passos
    - trocar a chave a ser removida com a sua chave sucessora imediata (que está em um nó folha)
    - remover a chave diretamente do nó folha
-



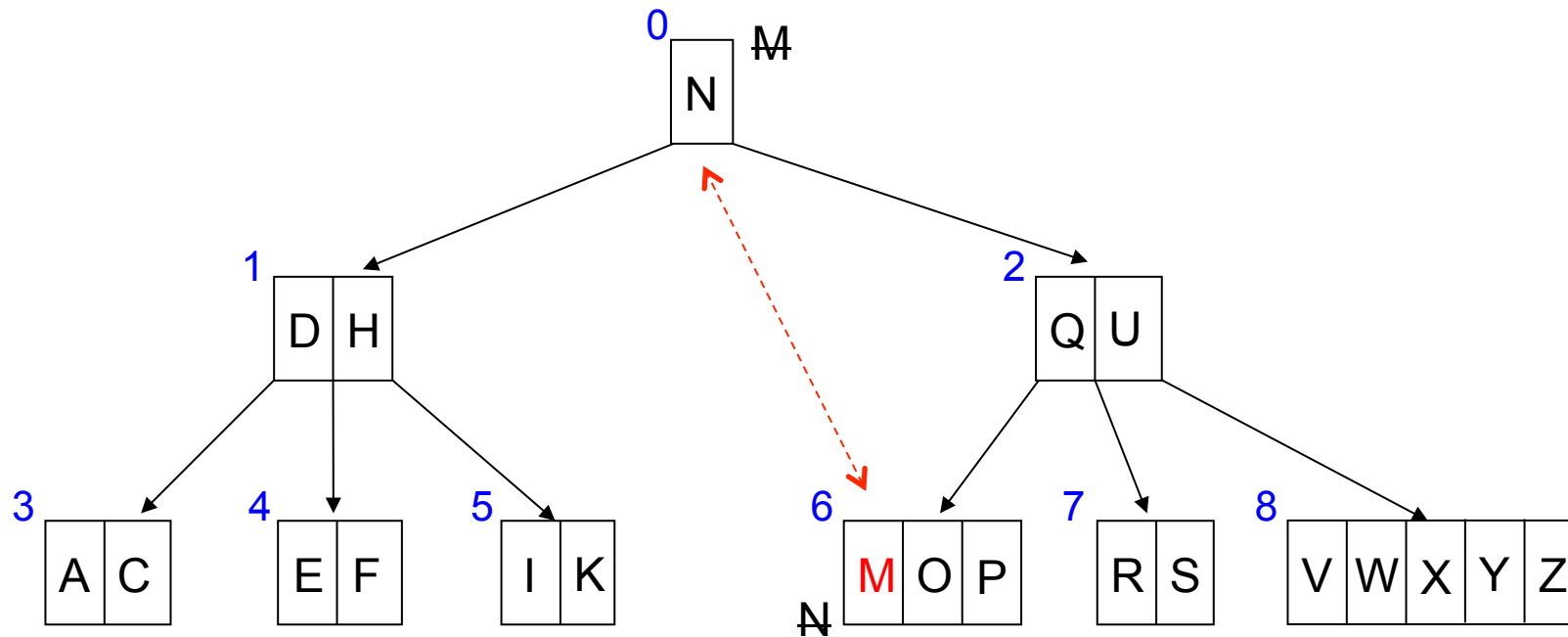
# Remoção: Caso 2 (1/3)

- Remoção de **M**



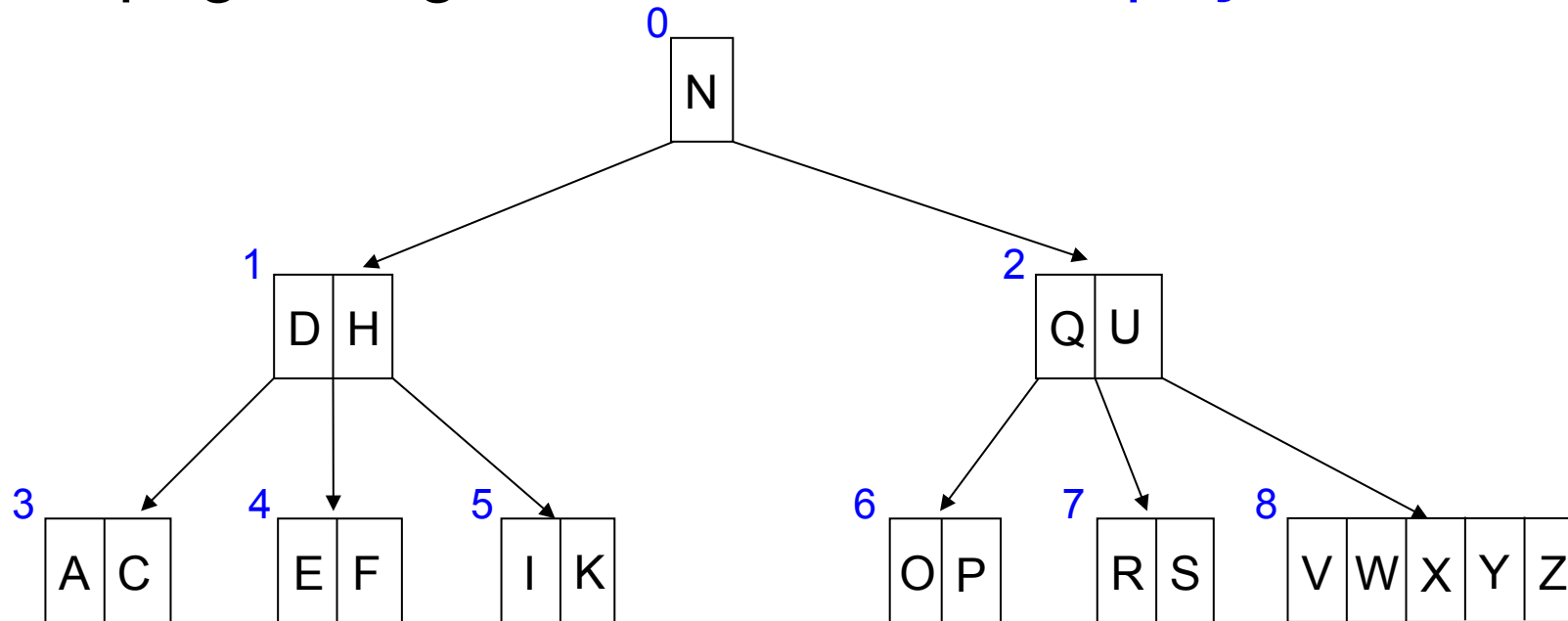
# Remoção: Caso 2 (2/3)

- Remoção de **M**  
– troca-se M com N



# Remoção: Caso 2 (3/3)

- Remoção de M
  - elimina-se M
  - página 6 garante a taxa de ocupação

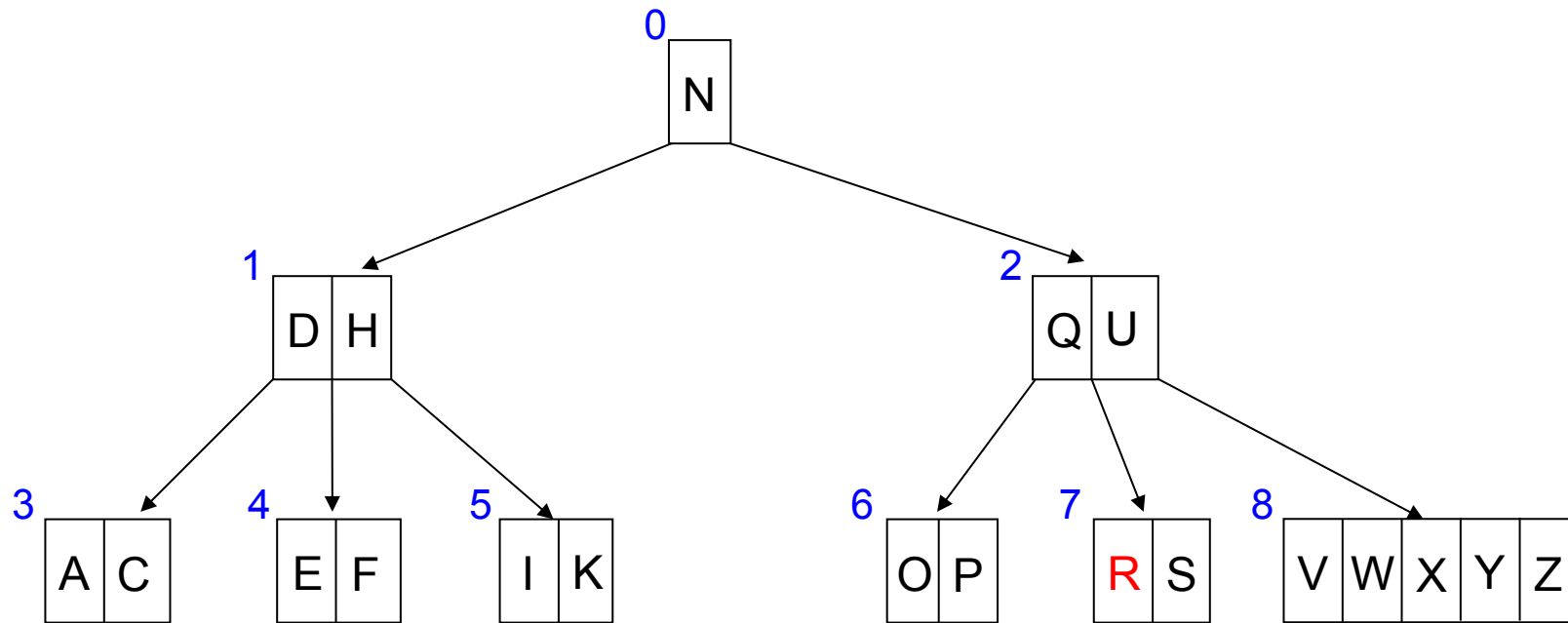


# Remoção: Caso 3

- Remoção de uma chave em um nó, causando *underflow*
  - Solução: Redistribuição
    - procurar uma página irmã (i.e., que possui o mesmo pai) adjacente que contenha mais chaves do que o mínimo
    - se encontrou
      - redistribuir as chaves entre as páginas
      - reacomodar a chave separadora, modificando o conteúdo do nó pai
-

# Remoção: Caso 3 (1/3)

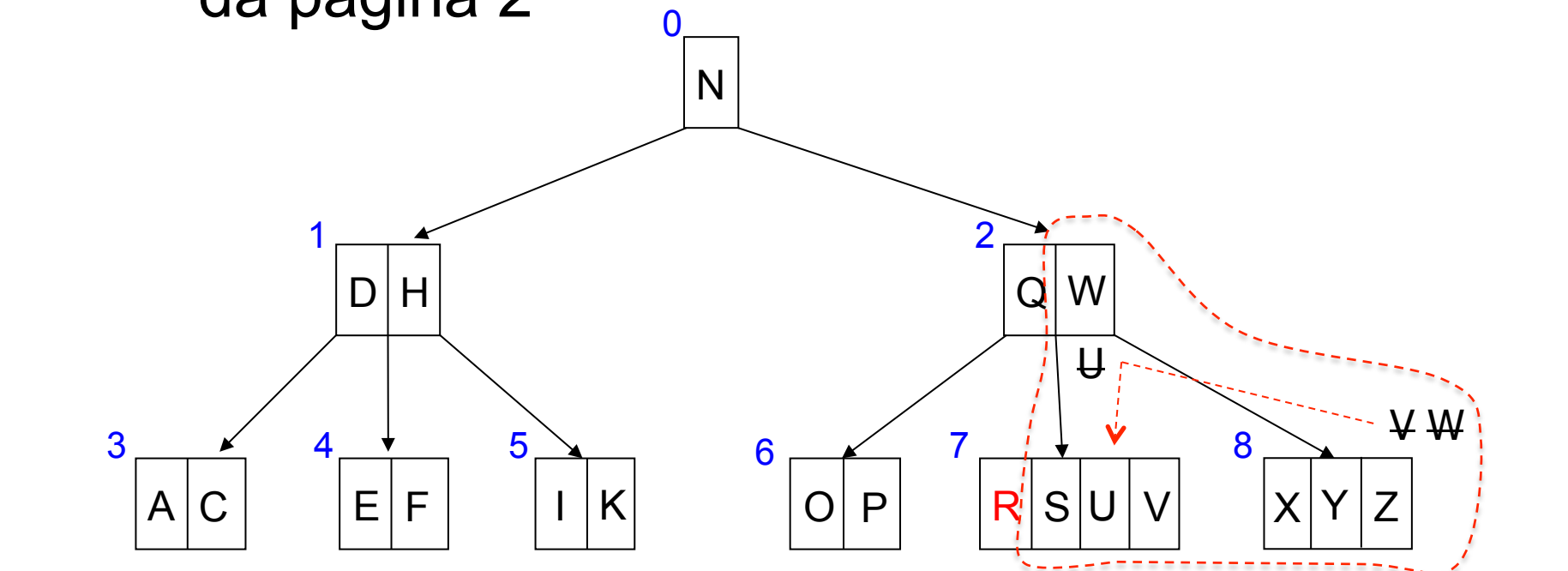
- Remoção de R



# Remoção: Caso 3 (2/3)

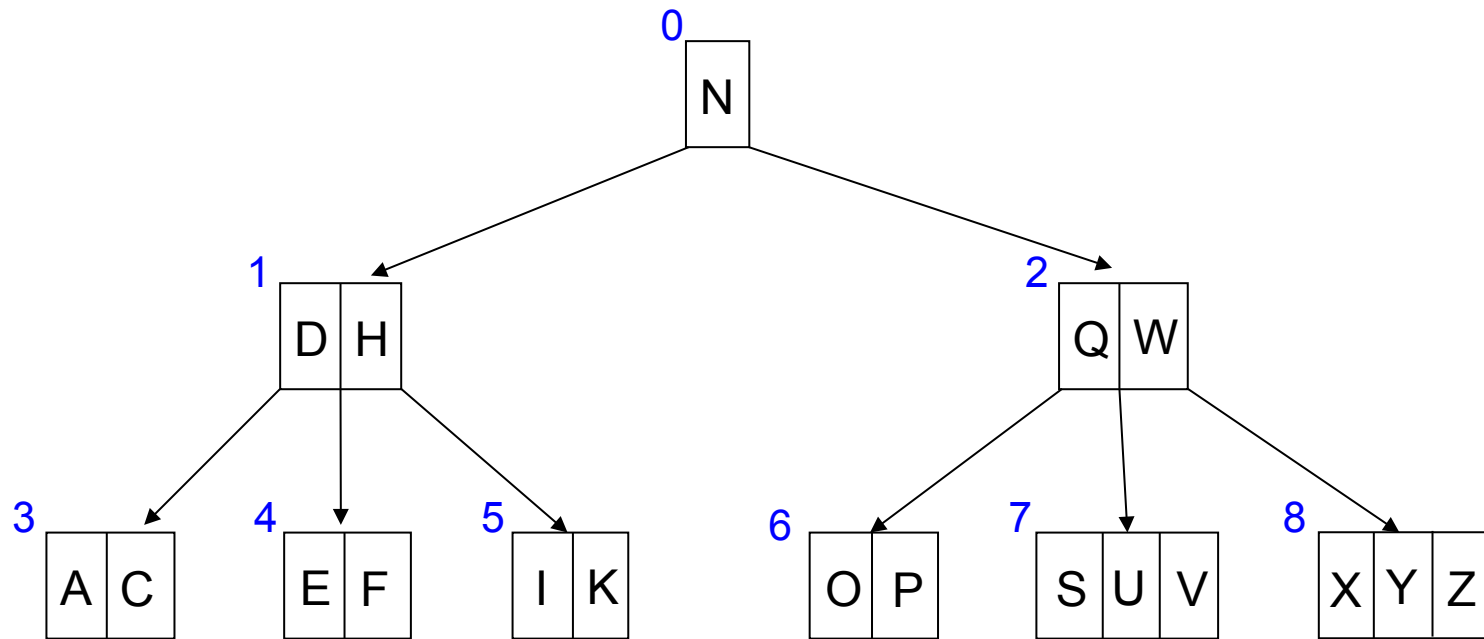
- Remoção de **R**

– para evitar *underflow* na página 7, redistribuíse as chaves entre as páginas 7 e 8 por meio da página 2



# Remoção: Caso 3 (3/3)

- Remoção de R
  - páginas 7 e 8 garantem a taxa de ocupação



# Remoção: Caso 4

- Remoção de uma chave em um nó, causando *underflow* e a redistribuição não pode ser aplicada
  - Solução: Concatenação
    - combinar para formar uma nova página
      - o conteúdo do nó que sofreu *underflow*
      - o conteúdo de um nó irmão adjacente
      - a chave separadora no nó pai
    - tratar o *underflow* no nó pai, caso necessário
-



# Concatenação

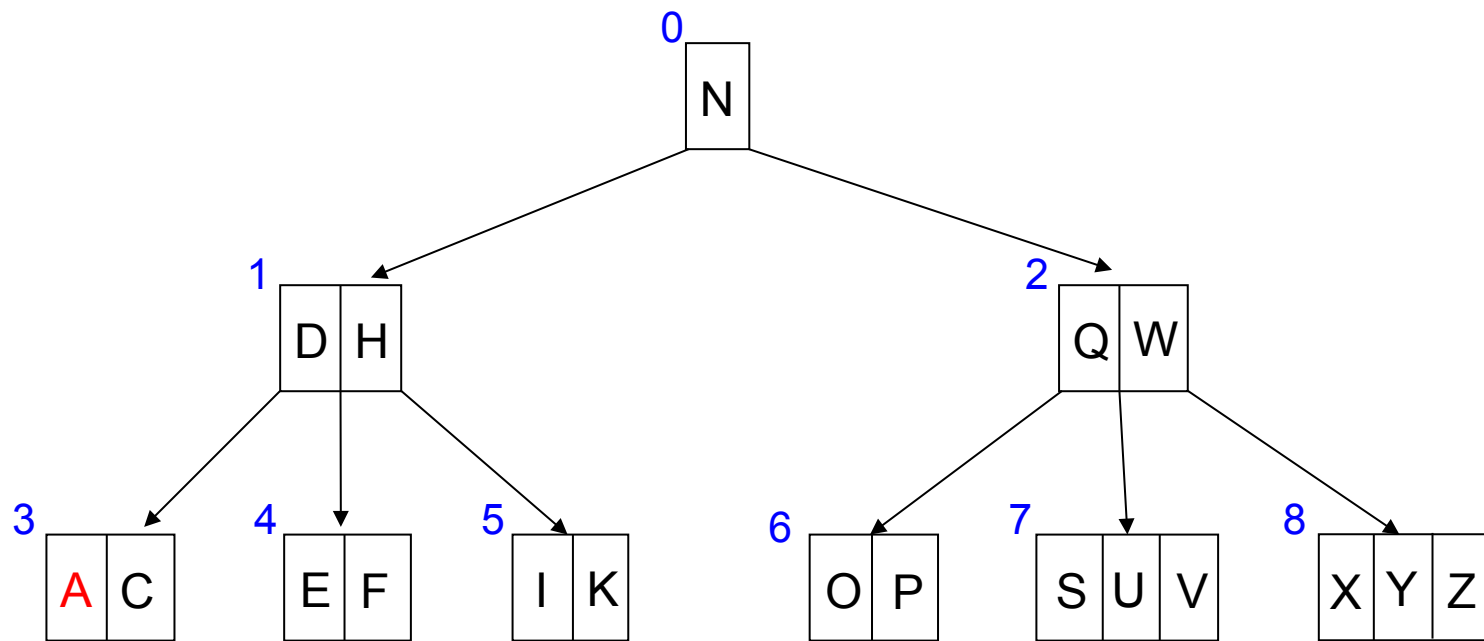
- Processo inverso do *split*
  - Características
    - reverte a promoção de uma chave
    - pode causar *underflow* no nó pai
- ⇒ concatenação pode ser propagada em direção ao nó raiz

ocorre a redução no número total  
de nós da árvore

---

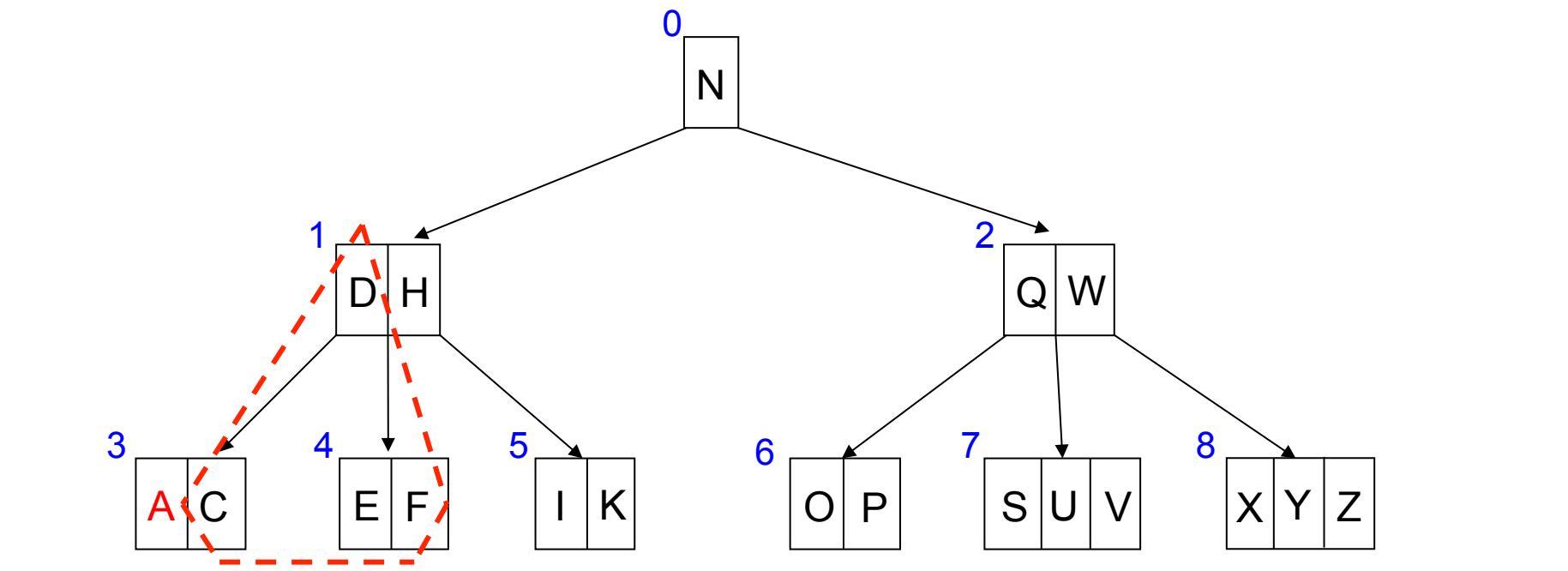
# Remoção: Caso 4 (1/3)

- Remoção de **A**



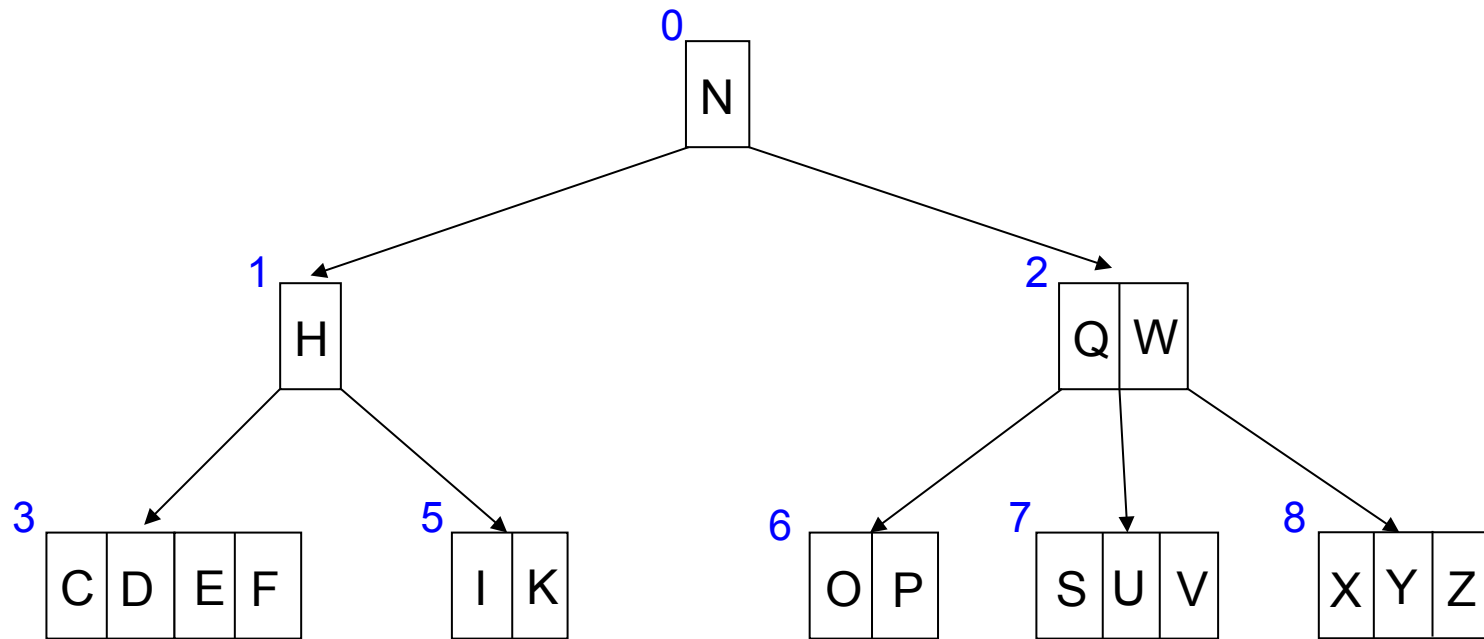
# Remoção: Caso 4 (2/3)

- Remoção de **A**
  - concatena-se as páginas 3 e 4 por meio da página 1



# Remoção: Caso 4 (3/3)

- Remoção de **A**
  - gera-se *underflow* na página 1, o qual precisa ser tratado

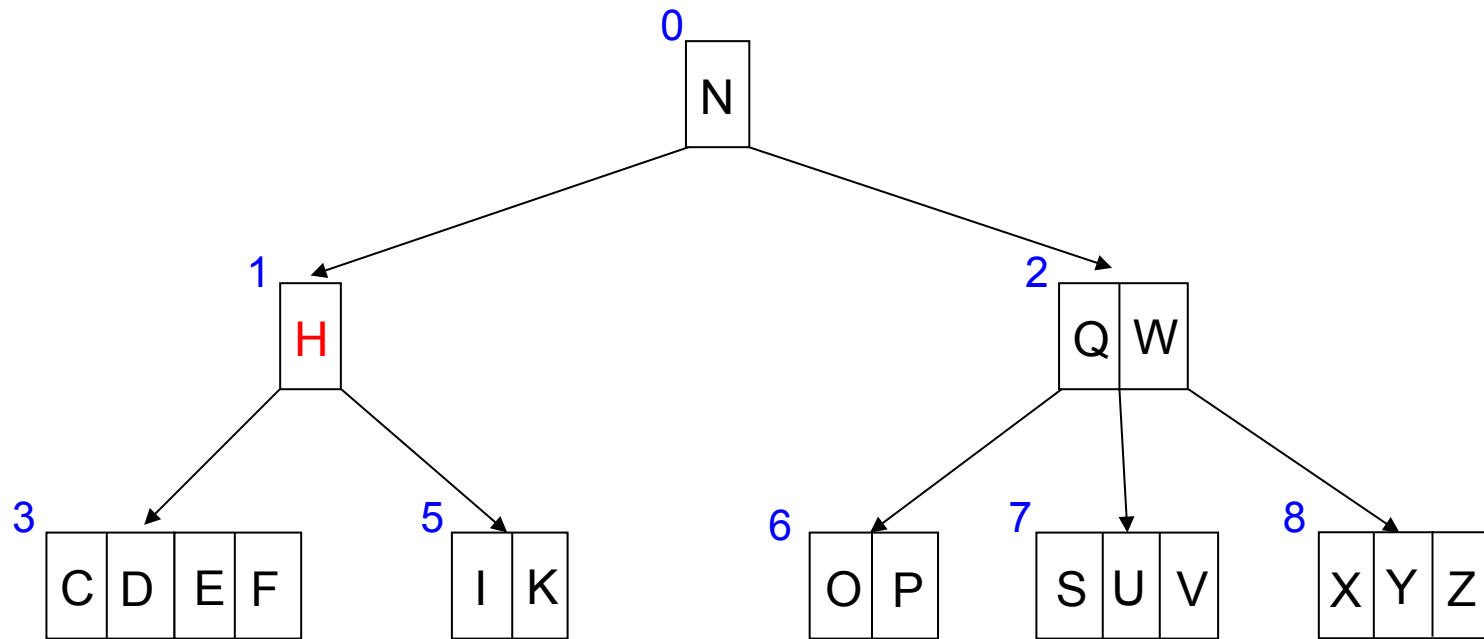


# Remoção: Caso 5

- *Underflow* no nó pai causado pela remoção de uma chave em um nó filho
  - Solução
    - utilizar *redistribuição* ou *concatenação*, dependendo da quantidade de chaves que a página irmã adjacente contém
-

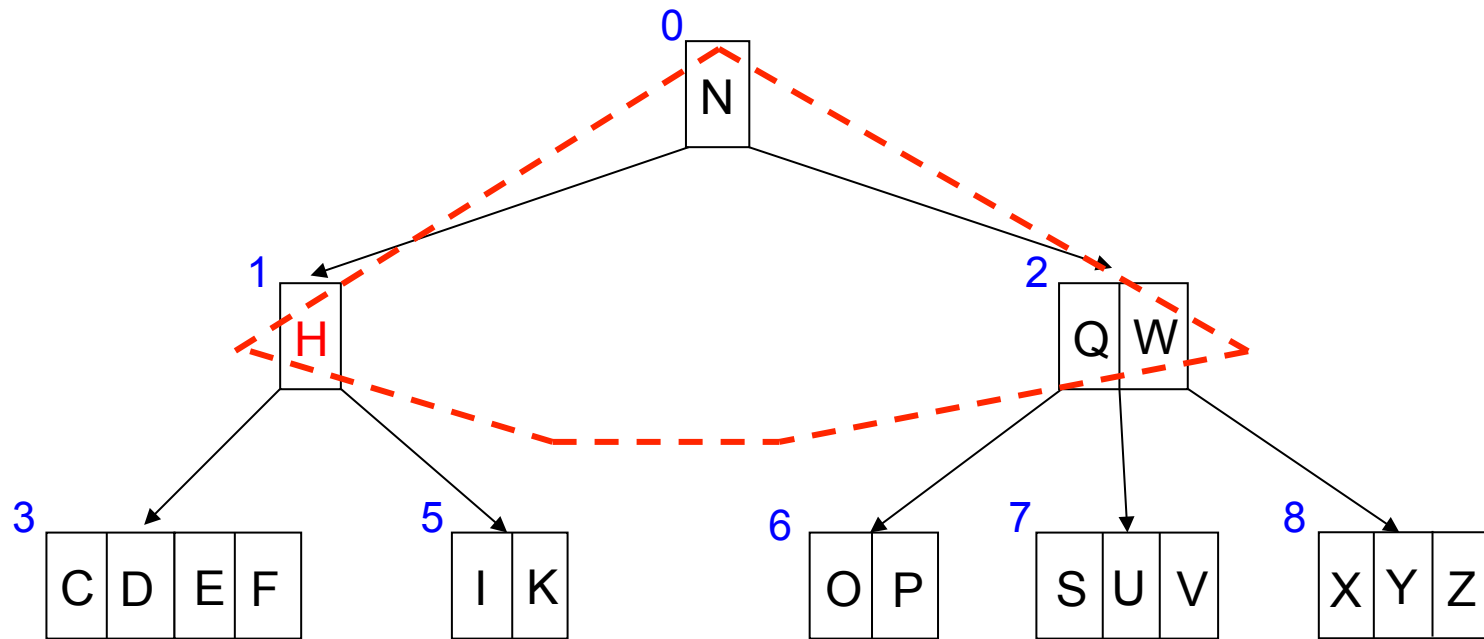
# Remoção: Caso 5 (1/3)

- Propagação do *underflow*



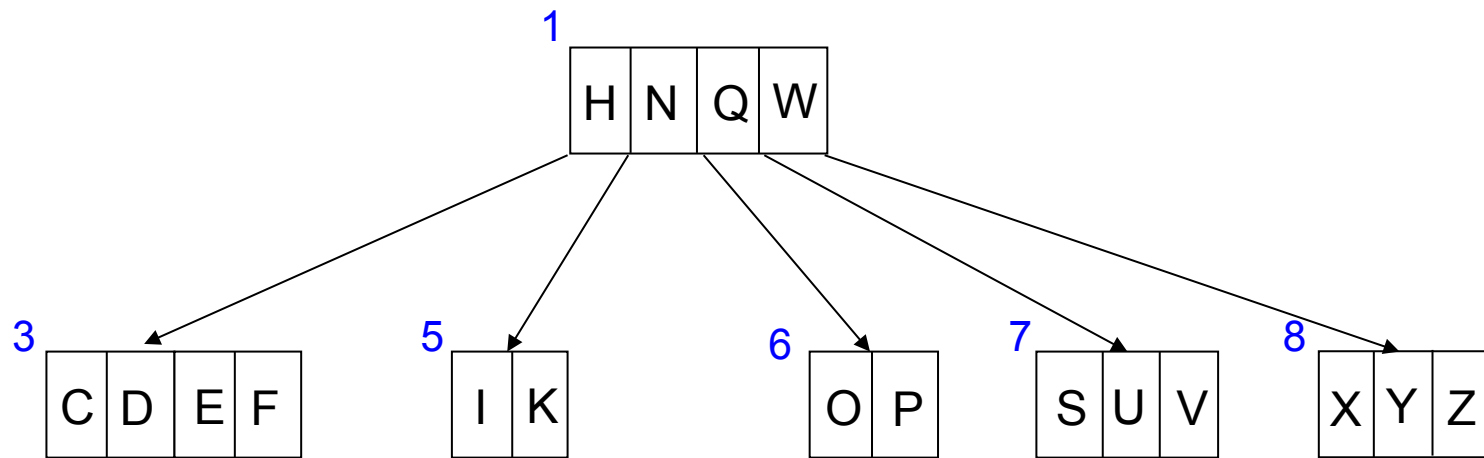
# Remoção: Caso 5 (2/3)

- Propagação do *underflow*
  - concatena-se as páginas 1 e 2 por meio da página 0



# Remoção: Caso 5 (3/3)

- Propagação do *underflow*





# Remoção: Caso 6

- Diminuição da altura da árvore
  - Característica
    - o nó raiz possui uma única chave
    - a chave é absorvida pela concatenação de seus nós filhos
  - Solução
    - eliminar a raiz antiga
    - tornar no nó resultante da concatenação dos nós filhos a nova raiz da árvore
-

# Algoritmo

1. se a chave a ser removida **não** estiver em um nó folha, **troque-a** com sua sucessora imediata, que está em um nó folha
  2. **remova** a chave
  3. após a remoção, se o nó satisfaz o número mínimo de chaves, nenhuma ação adicional é requerida
-

# Algoritmo

4. após a remoção, caso ocorra *underflow*, verifique o número de chaves nos nós irmãos adjacentes à esquerda e à direita
    - a. se algum nó irmão adjacente possui mais do que o número mínimo de chaves, aplique a *redistribuição*
    - b. se nenhum nó irmão adjacente possui mais do que o número mínimo de chaves, aplique a *concatenação*
-

# Algoritmo

5. se ocorreu **concatenação**, repita os passos 3 a 5 para o nó pai
  6. se a última chave da raiz for removida, a altura da árvore é **diminuída**
-

# Redistribuição

- Representa uma idéia inovadora
    - diferente do *split* ou da concatenação
  - Não se propaga para os nós superiores
    - apenas efeito local na árvore
  - Baseada no conceito de nós irmãos adjacentes
    - dois nós logicamente adjacentes, mas com pais diferentes não são irmãos
-

# Redistribuição

- Não fixa a forma na qual as chaves devem ser redistribuídas
  - possibilidade 1: mover somente uma chave, mesmo que a distribuição das chaves entre as páginas não seja uniforme
  - possibilidade 2: mover k chaves
  - possibilidade 3: distribuição uniforme das chaves entre os nós



mais  
comum

---