

Capítulo 8

Árvores B

Parte 3



Programa

- Introdução
- Operações básicas sobre arquivos
- Armazenagem secundária
- Conceitos básicos sobre estrutura de arquivo
- Organizando arquivos para desempenho
- Indexação
- Processamento co-seqüencial e ordenação
- **B-Tree e outras organizações em árvores**
- B+Tree e acesso seqüencial indexado
- Hashing
- Hashing estendido



Árvore-B

- Operações Básicas:
 - Inserções
 - Divisão (*Splitting*)
 - Promoção
 - Eliminação

● ● ● | Árvore-B

- Inserção:
 - A inserção ocorre normalmente enquanto existe espaço dentro de uma página:
 - Mantenha as chaves ordenadas.
 - Inserção das chaves C e D, no início:

*	C	*	D	*		*		*		*		*		*
---	---	---	---	---	--	---	--	---	--	---	--	---	--	---

- Inserção das chaves G, A, F, B, E:

*	A	*	B	*	C	*	D	*	E	*	F	*	G	*
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

As chaves são copiadas para a memória onde as inclusões ocorrem.



Árvore-B

- Inserir uma nova chave numa folha cheia
 - Ocorre uma divisão que consiste na criação de uma nova página, dividindo-se as chaves existentes entre as páginas.
 - A chave que é o melhor separador é promovida para o nível superior.

Um bom separador divide as chaves de forma uniforme.

Árvore-B

- Assim, inserir a chave J na página inicial

*	A	*	B	*	C	*	D	*	E	*	F	*	G	*
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

- divide a página em duas outras:

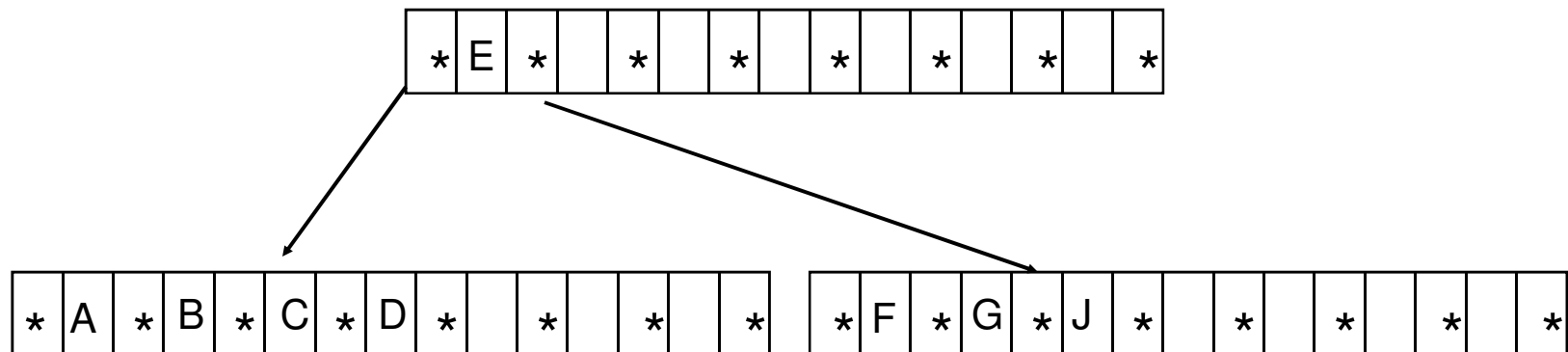
*	A	*	B	*	C	*	D	*		*		*		*
---	---	---	---	---	---	---	---	---	--	---	--	---	--	---

*	F	*	G	*	J	*		*		*		*		*
---	---	---	---	---	---	---	--	---	--	---	--	---	--	---

- e promove a chave E (sobe um nível).

Árvore-B

- Para manter a estrutura de árvore, cria-se uma nova raiz:





Árvore-B

- Exemplo: inserção das chaves

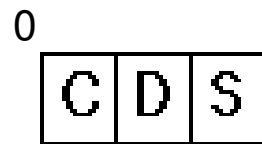
C	S	D	T	A	M	P	I	B	W	N	G	U	R	K	E	H	O	L	J	Y	Q	Z	F	X	V
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

- em uma árvore-B de ordem 4

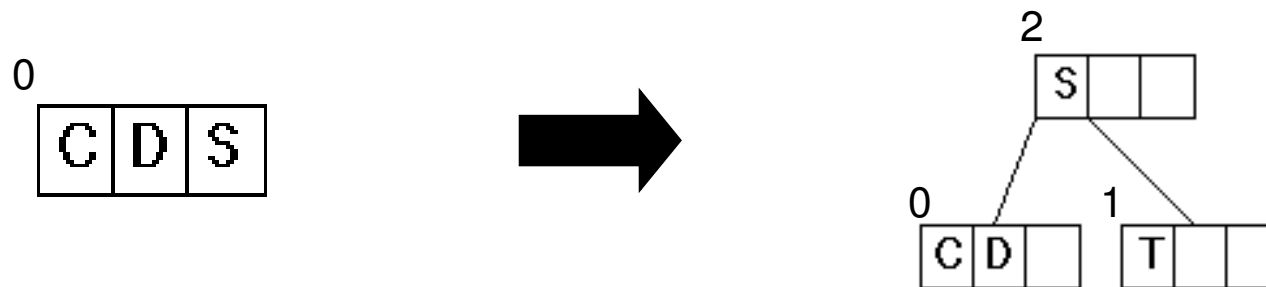
3 chaves e 4 ponteiros por chave.
A árvore é contruída de forma bottom-up.
As páginas são de tamanho fixo.

Árvore-B

- Inserção de C S D dentro da página inicial:



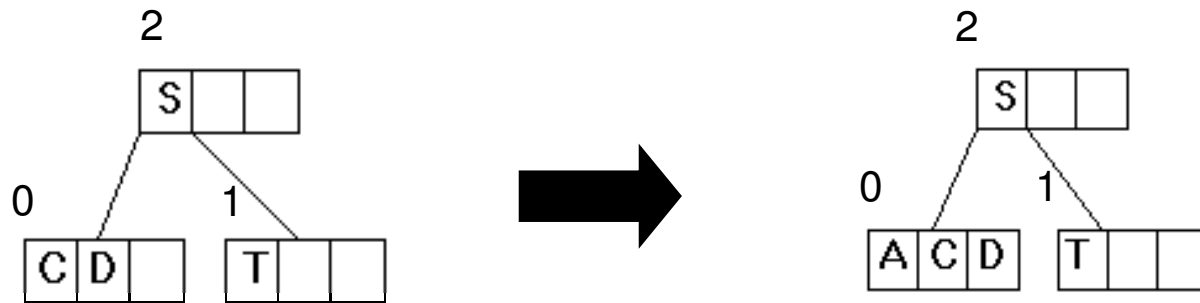
- Inserção de T força a divisão/promoção de S:



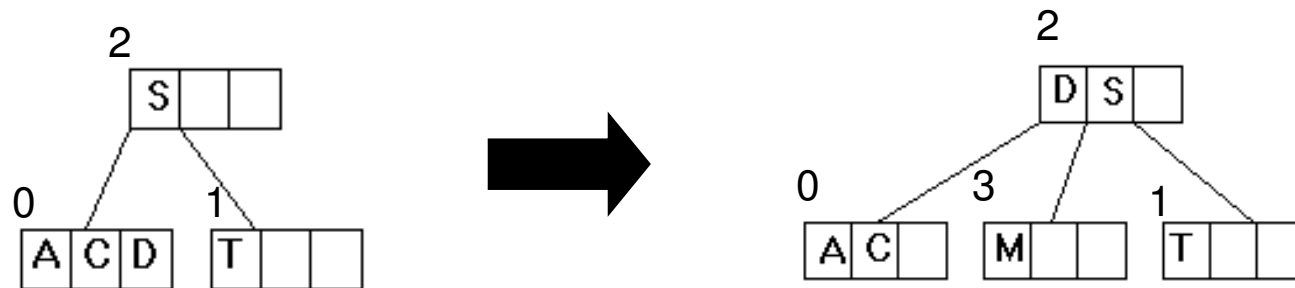
C	S	D	T	-	A	M	P	I	B	W	N	G	U	R	K	E	H	O	L	J	Y	Q	Z	F	X	V
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Árvore-B

- Inserção de A



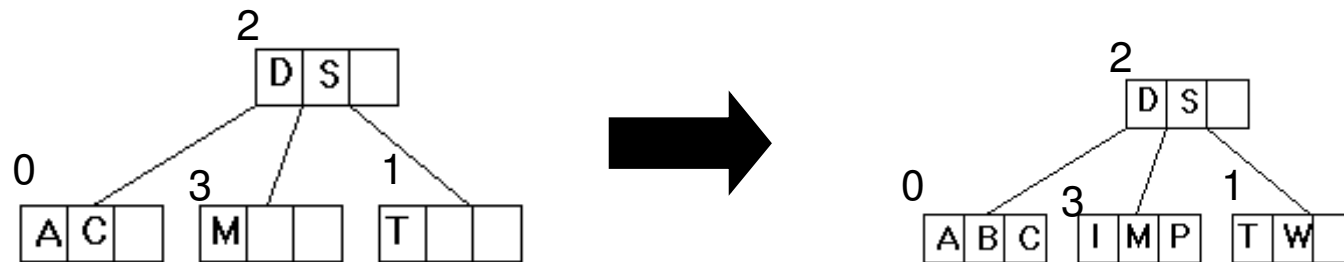
- Inserção de M força outra divisão e promoção de D (sem gerar um nó novo raiz)



C S D T A M - P I B W N G U R K E H O L J Y Q Z F X V

Árvore-B

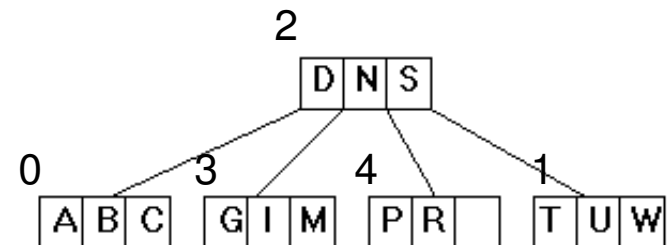
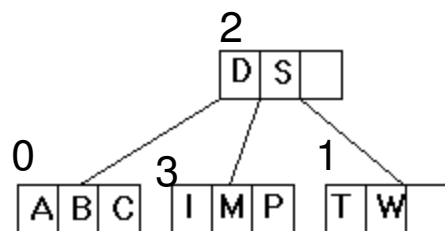
- Inserção de P, I, B e W nas páginas existentes



C	S	D	T	A	M	P	I	B	W	-	N	G	U	R	K	E	H	O	L	J	Y	Q	Z	F	X	V
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Árvore-B

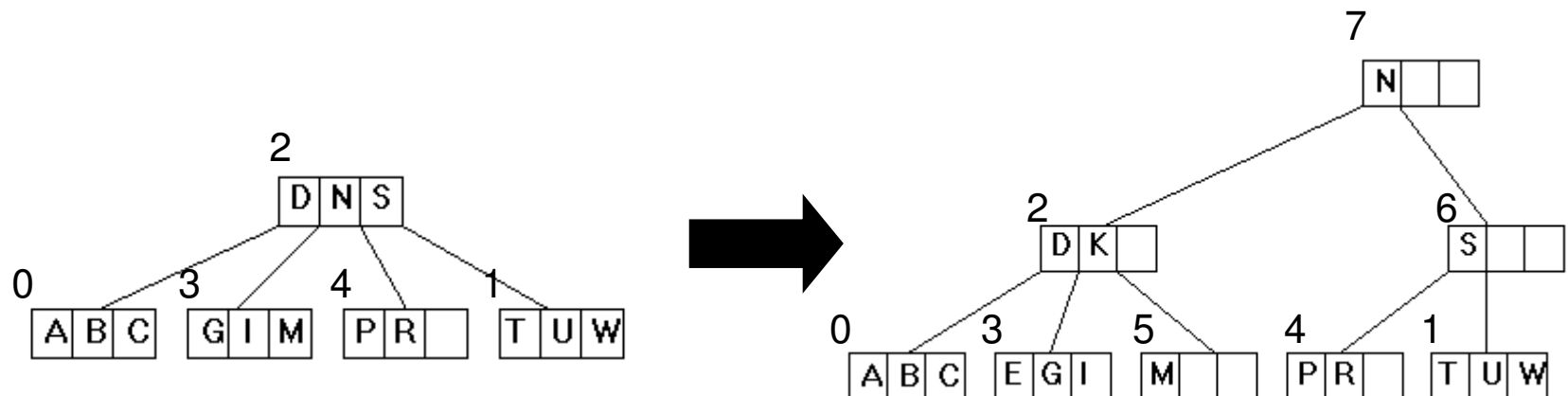
- Inserção de N precisa de outra divisão seguida de promoção de N
 - Inserção de G, U e R, são feitas a seguir.



C S D T A M P I B W N G U R - K E H O L J Y Q Z F X V

Árvore-B

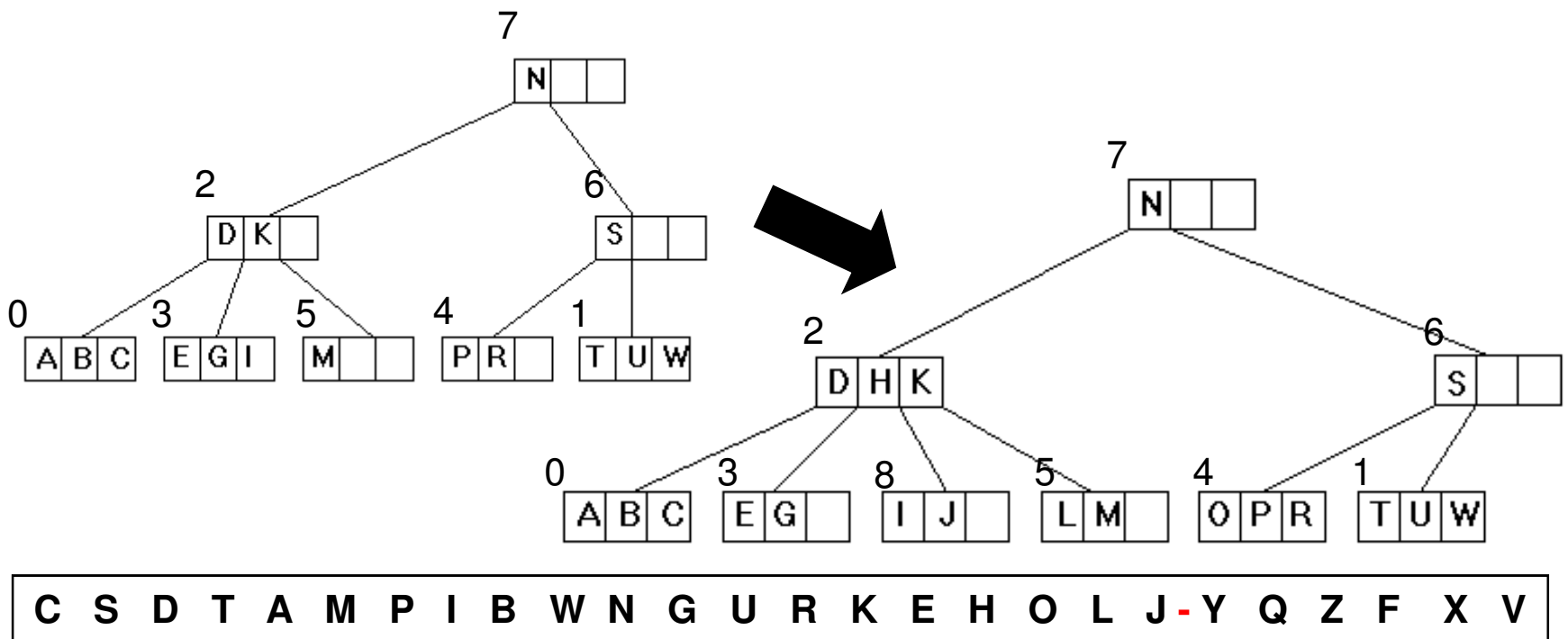
- Inserção de K resulta na divisão e promoção de K, que por sua vez força a divisão e promoção do N. Inserção de E é normal.



C S D T A M P I B W N G U R K E - H O L J Y Q Z F X V

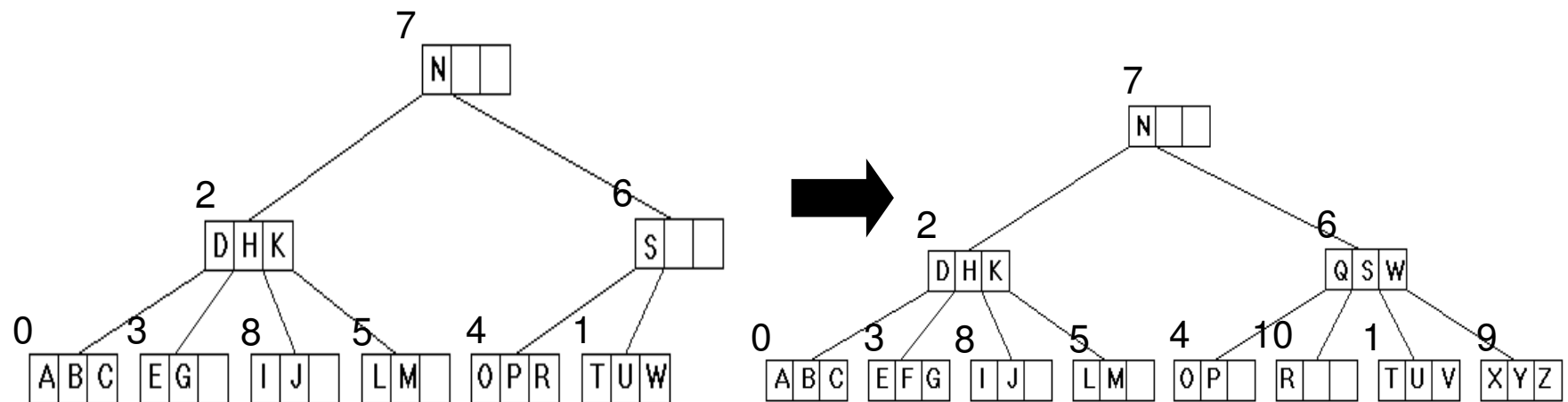
Árvore-B

- Inserção de H resulta em divisão do nó folha. H é promovido. O, L e J são inseridos a seguir.



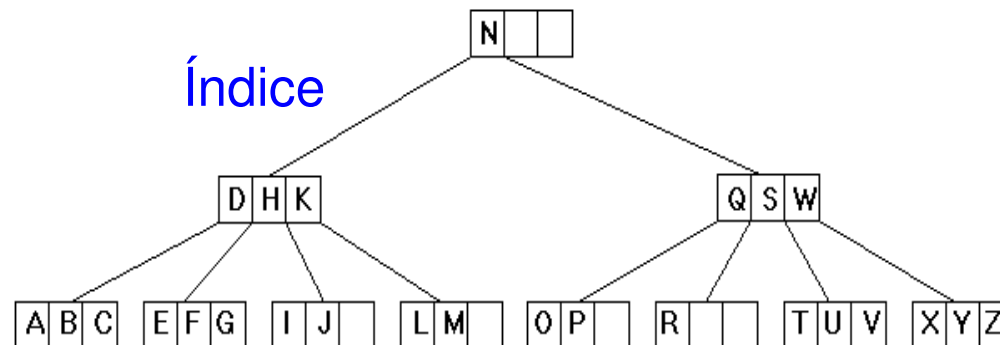
Árvore-B

- Inserção de Y e Q forçam mais duas divisões nos nós folhas. As letras restantes são adicionadas



C S D T A M P I B W N G U R K E H O L J Y Q Z F X V -

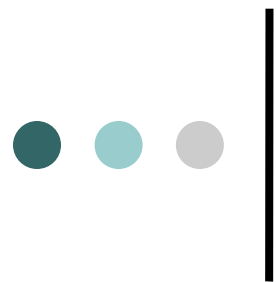
Árvore-B como índice



Arquivo
Básico

C	...	S	...	D	...	T	...	A	...	M	...	P	...
I	...	B	...	W	...	N	...	G	...	U	...	R	...
K	...	E	...	H	...	O	...	L	...	J	...	Y	...
Q	...	Z	...	F	...	X	...	V	...				

Temporal = desordenado



Árvore-B

- Inserção, Splitting (divisão), e Promoção:
 - Começa com uma busca que procede **top-down** para o nível das folhas, buscando a chave ou a folha para inserí-la (se a chave já estiver na árvore retorna msg de erro/chave duplicada).
 - Depois de encontrar o local de inserção no nível das folhas, o trabalho de inserção, divisão, e promoção procede de **baixo para cima**, no mesmo caminho feito durante a busca.



Árvore-B

- Algoritmos de Busca e Inserção estão detalhados na seção 8.9 do livro texto.



Árvore-B

- Ordem m : número máximo de descendentes de uma página (Knuth).
- PROPRIEDADES:
 1. Uma página possui no máximo m (páginas) descendentes.
 2. Toda página, exceto a raiz e as folhas, tem no mínimo $\lceil m/2 \rceil$ (páginas) descendentes.
 3. A raiz tem no mínimo 2 descendentes, a menos que seja uma folha.
 4. Todas as folhas aparecem no mesmo nível.
 5. Uma página que não é folha e possui k (páginas) descendentes, contém $k-1$ chaves.
 6. Uma página folha contém no mínimo $\lceil m/2 \rceil - 1$ e no máximo $m-1$ chaves.



Árvore-B

- Essas propriedades garantem um número mínimo de chaves em cada página e a escolha correta das chaves da raiz.
- Isso elimina os problemas encontrados na paginação de árvores construídas com a abordagem top-down.



Árvore-B

- É importante um entendimento quantitativo da relação entre:
 - O tamanho da página da árvore-B.
 - O número de chaves a serem armazenadas na árvore.
 - O número de níveis que a árvore pode ser expandida.

Por exemplo:

- Para armazenar 1.000.000 de chaves
- Usando uma árvore-B de ordem 512 (máx 511 chaves por página)

No pior caso, qual o número máximo de acessos a disco necessário para localizar uma chave na árvore? (É o mesmo que perguntar quão profunda será a árvore.)

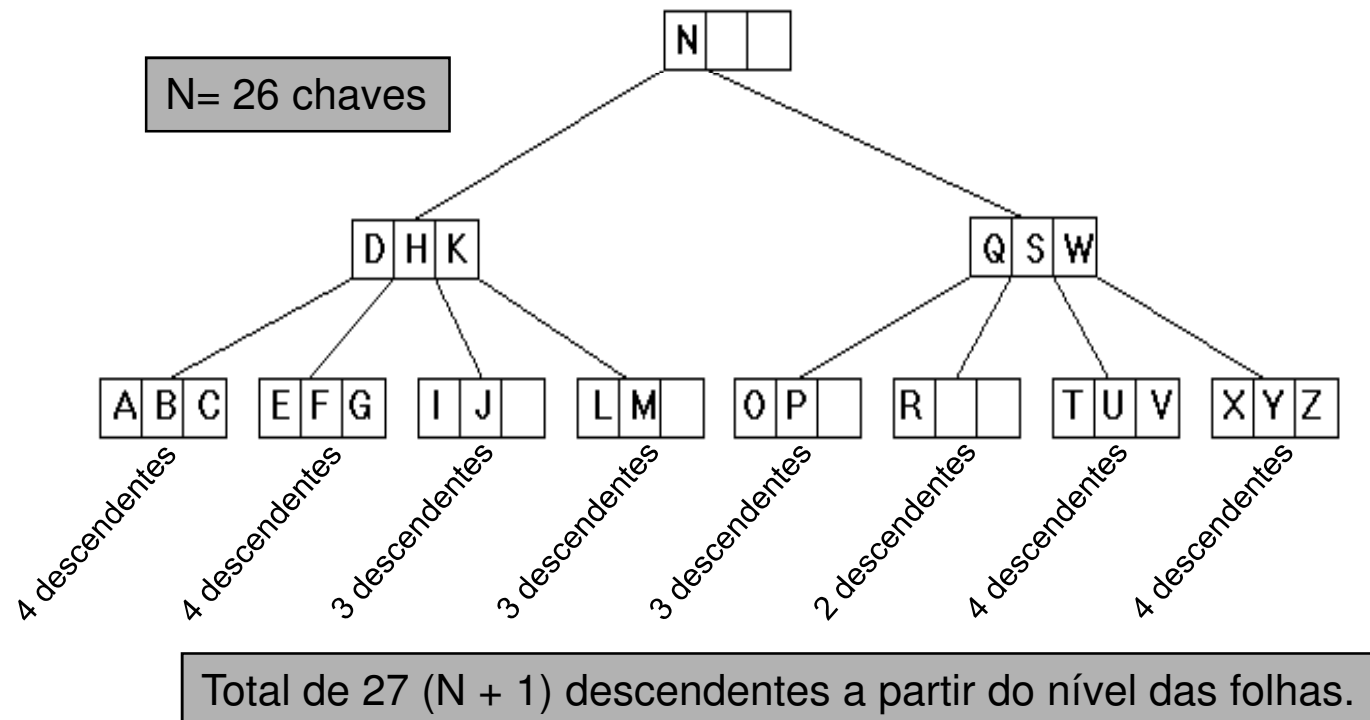


Árvore-B

- Profundidade da Árvore:
 - A profundidade da árvore está diretamente relacionada com o número máximo de seeks para encontrar uma determinada chave.
 - Os fatores que influenciam a profundidade da árvore são:
 - fator de ramificação (ordem da árvore)
 - número efetivo de chaves em cada página
 - quantidade de chaves (total)
 - política de carga mínima (quando dividir a página)

Árvore-B

- O número de descendentes em qualquer nível da árvore-B é o número de chaves contidas naquele nível e níveis acima + 1.



Árvore-B

- O pior caso de profundidade da árvore:
 - Ocorre quando cada página da árvore (nó) tem o número mínimo de filhos (descendentes).

Level	Minimum number of keys (children)
1 (root)	2
2	$2 \cdot \lceil m/2 \rceil$
3	$2 \cdot \lceil m/2 \rceil \cdot \lceil m/2 \rceil = 2 \cdot \lceil m/2 \rceil^2$
4	$2 \cdot \lceil m/2 \rceil^3$
...	...
d	$2 \cdot \lceil m/2 \rceil^{d-1}$

● ● ● | Árvore-B

- Sabemos que uma árvore com N chaves tem $N + 1$ descendentes a partir do seu nível das folhas.
- Vamos chamar a profundidade da árvore no nível das folhas de d .
- Então, se nós temos um total de N chaves até o nível d :

$$N + 1 \geq 2 \cdot \left\lceil m / 2 \right\rceil^{d-1}$$

Árvore-B

$$N + 1 \geq 2 \cdot \lceil m/2 \rceil^{d-1}$$



$$d \leq 1 + \log_{\lceil m/2 \rceil} ((N + 1) / 2)$$

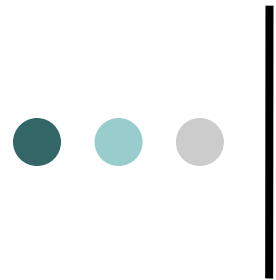
Esta expressão nos dá um limite para a profundidade da árvore-B com N chaves.

- Para $N = 1.000.000$ e ordem $m = 512$, temos

$$d \leq 1 + \log_{256} 500,000$$

$$d \leq 3.37$$

Existem no máximo 3 níveis numa árvore-B de ordem 512 contendo 1.000.000 de chaves.

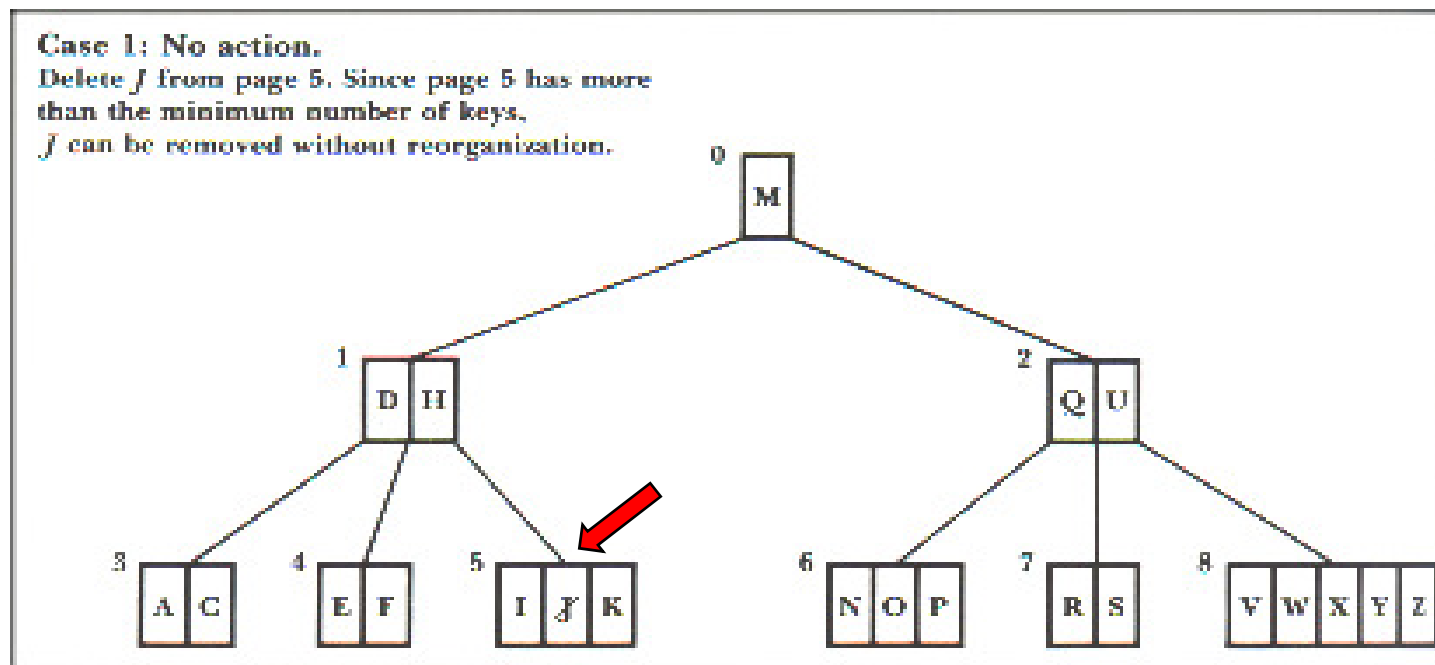


Árvore-B (Eliminação)

- Eliminação de chaves
 - Redistribuição e Concatenação
 - As propriedades da Arvore-B, descritas antes, garantem que haja um número mínimo de elementos em uma página.
 - Precisamos manter esta propriedade no processo de eliminação de uma chave

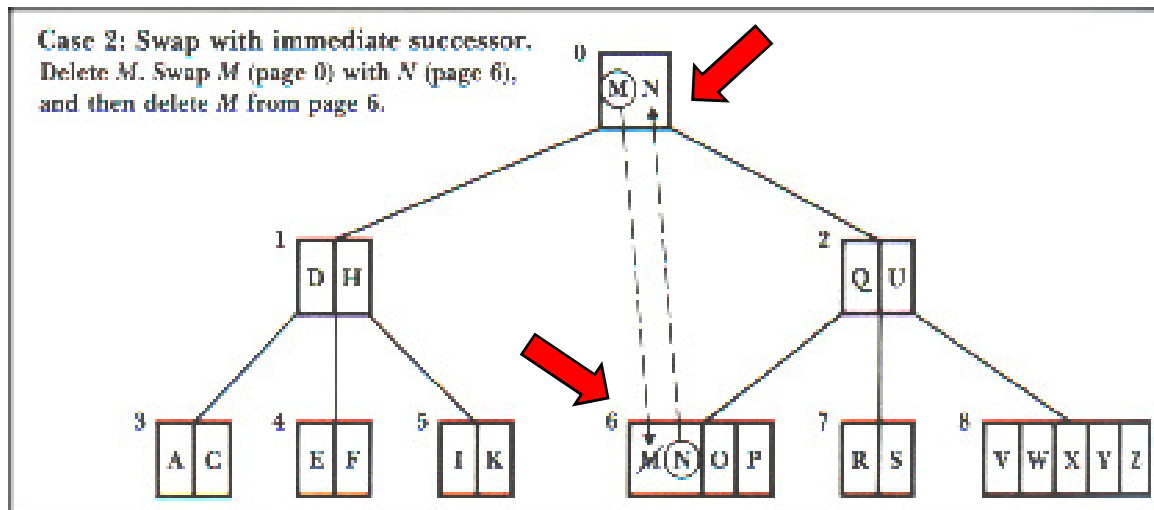
Árvore-B (Eliminação Simples)

- **Caso 1:** a chave está numa folha, e após a eliminação ainda fica um número mínimo de chaves $m=6$ e $\lceil m/2 \rceil - 1 = 2$
- Solução: chave J é retirada e registros internos à página são reordenados.



Árvore-B (Troca e Eliminação)

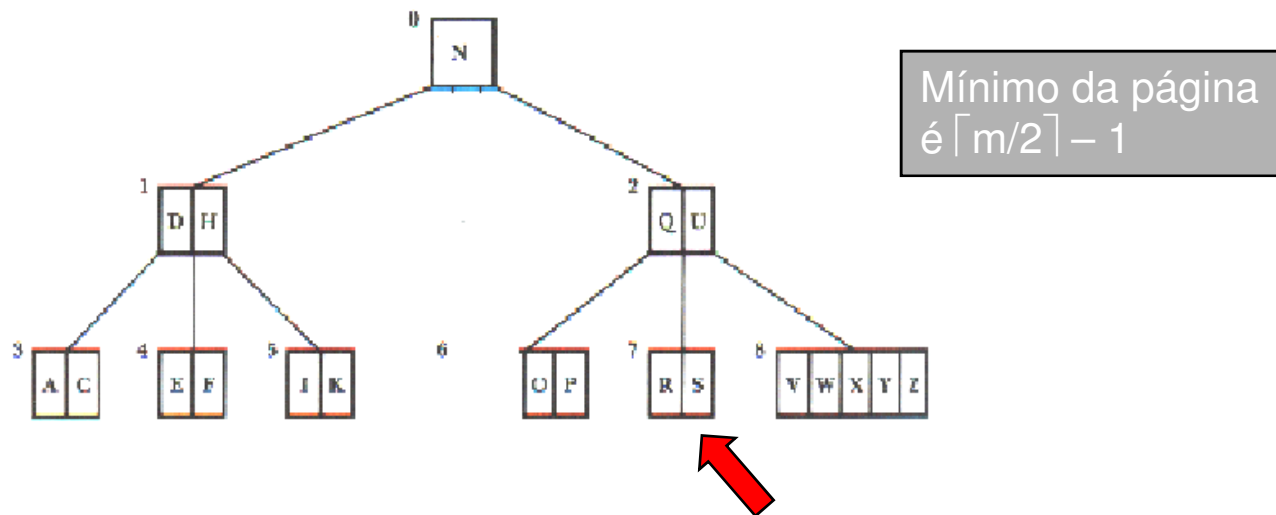
- **Caso 2:** eliminação de uma chave que não está na folha.
- Solução: trocar a chave M com seu sucessor imediato N que está numa folha e eliminar a chave.



Somente excluimos
chaves das folhas!!!

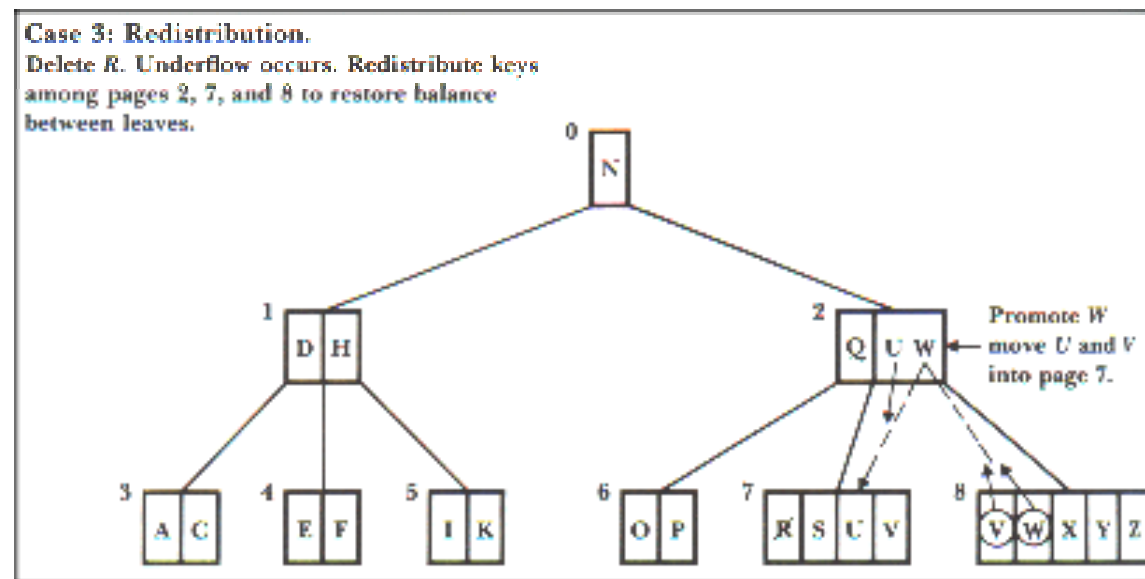
Árvore-B (Eliminação e Redistribuição)

- **Caso 3:** a eliminação de R viola o mínimo e causa **underflow** na página 7.
- Solução: redistribuição com uma página irmã.



Árvore-B (Eliminação e Redistribuição)

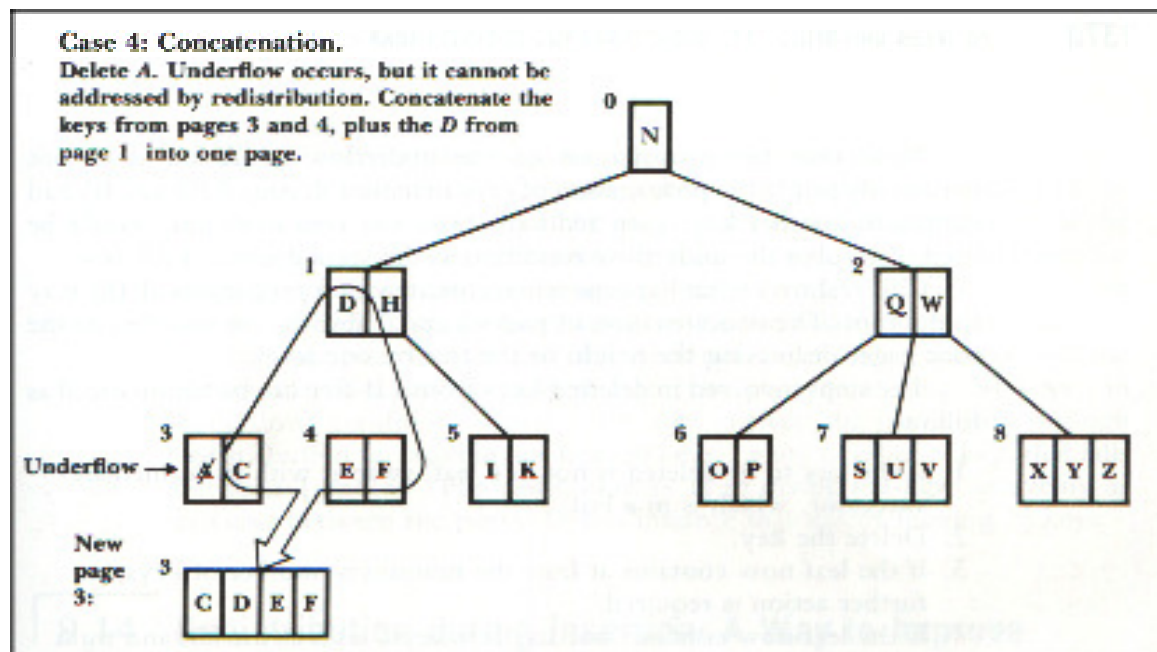
- **Caso 3:** (continuação...)



Mínimo da página é $\lceil m/2 \rceil - 1$

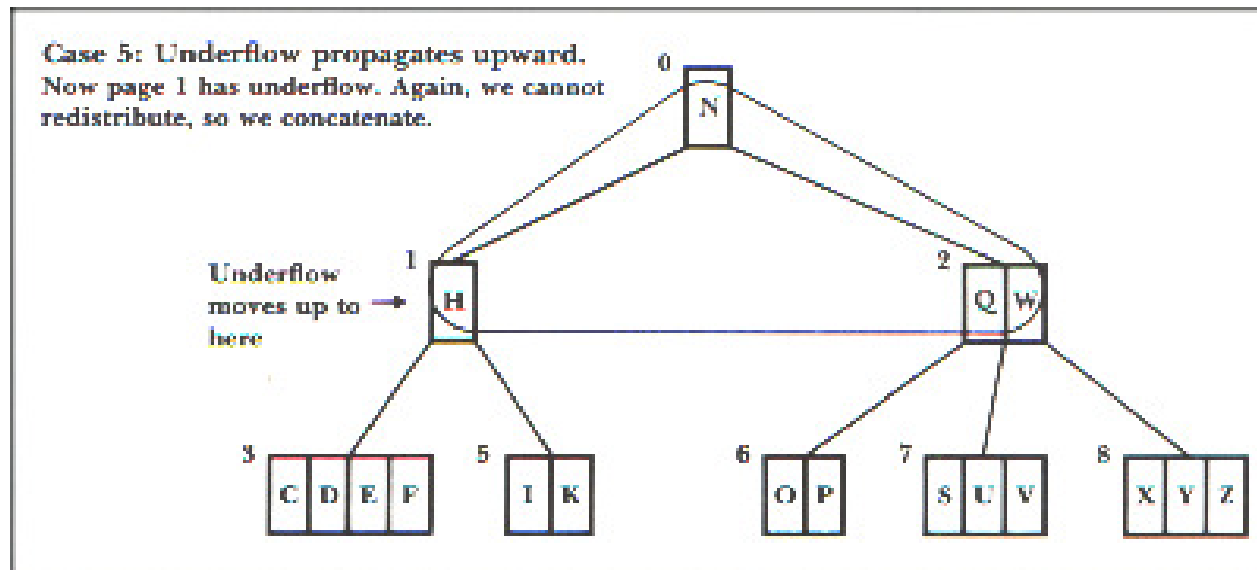
Árvore-B (Elimina e Concatena)

- **Caso 4:** ocorre o underflow e a redistribuição não pode ser aplicada com as vizinhas.
- Solução: concatenação



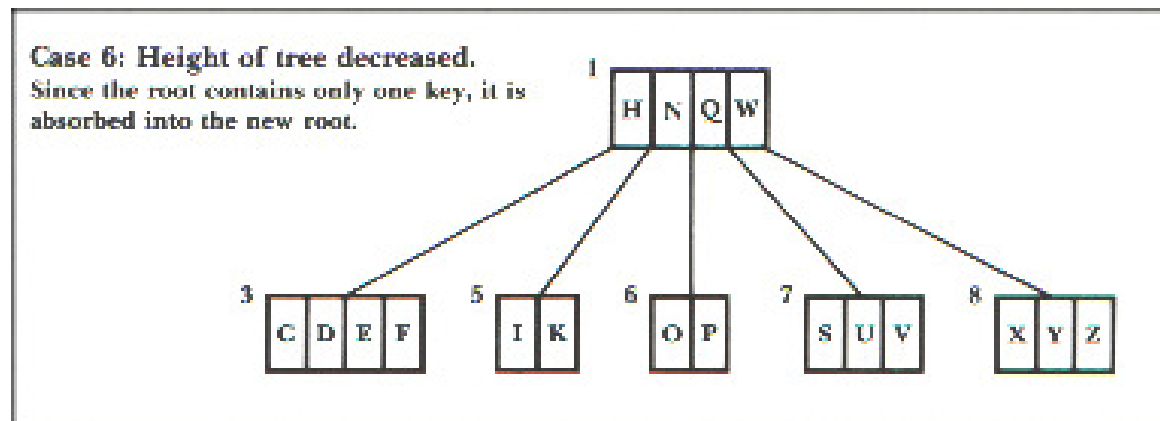
Árvore-B (Eliminação propaga para o pai)

- **Caso 5:** underflow na página pai.
- Solução: tenta-se a redistribuição. Se não for possível, utiliza-se a concatenação



● ● ● | Árvore-B (Eliminação diminui a altura)

- **Caso 6:** diminuição na altura da árvore
- Consequência da concatenação dos filhos do nó raiz





Árvore-B

- Visão Geral do Algoritmo de Exclusão:

1. Se a chave não estiver numa folha, troque-a com seu sucessor imediato.
2. Elimine a chave da folha.
3. Se a folha continuar com o número mínimo de chaves, fim.
4. A folha viola o mínimo. Verifique as páginas irmãs da esquerda e direita:
 - 4.1 se uma delas tiver mais que o número mínimo de chaves, aplique redistribuição.
 - 4.2 senão concatene a página com uma das irmãs e a chave pai.
5. Se ocorreu concatenação, aplique passos de 3 a 6 para a página pai.
6. Se a última chave da raiz for removida, a altura da árvore diminui.



Árvore-B

- Redistribuição Durante Inserção
 - A redistribuição é uma nova idéia (que é o contrário do Splitting).
 - Tem efeito apenas local.
 - É entre páginas irmãs e o pai.
 - O balanceamento de páginas
 - Pode ser atingido movendo-se apenas uma chave para a página onde o underflow ocorreu.
 - Mas, normalmente move-se mais do que uma.



Árvore-B

- Redistribuição Durante Inserção
 - Melhorar o uso de espaço.
 - Pode ser usada durante a inserção, como uma maneira de postergar a criação de páginas (por splitting).
 - Utilizando somente Splitting na inserção:
 - Normalmente o pior caso de uso de espaço é de cerca de 50%.
 - A média teórica de uso de espaço é cerca de 69% (Yao, 1978).
 - Experimentos mostram que com a redistribuição a utilização de espaço aumentou para 86% [(Davis, 1974) e (Crotzer, 1975)].

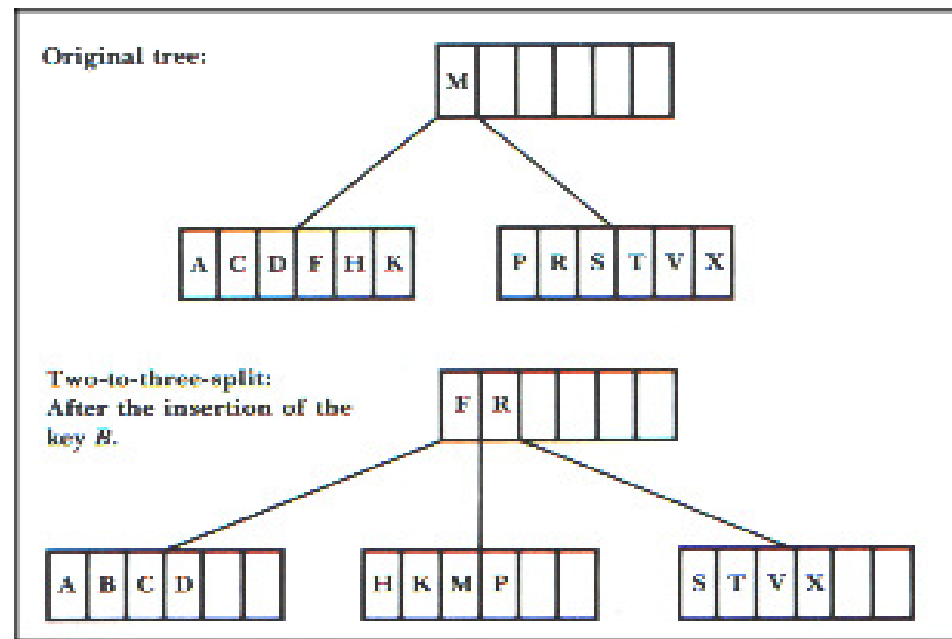


Árvore-B*

- Proposta por Donald Knuth em 1973
 - As Árvores-B* visam melhor aproveitamento do espaço, fazendo redistribuição durante a inserção e incluindo novas regras para o splitting (divisão).
- A divisão (splitting)
 - Somente ocorre quando duas páginas irmãs estiverem cheias.
 - Dessa forma, as três páginas resultantes da divisão terão em torno de 2/3 de ocupação, cada uma.

Árvore-B*

- Exemplo de divisão em Árvore-B*



ABCD F HKMP R STVX



Árvore-B*

- Propriedades:

1. Toda página tem no máximo m descendentes.



2. Toda página, exceto a raiz e as folhas, têm ao menos $(2m-1)/3$ descendentes.

3. A raiz tem pelo menos 2 descendentes (a menos que seja uma folha).

4. Todas as folhas estão no mesmo nível

5. Uma página não folha com k descendentes tem $k-1$ chaves.

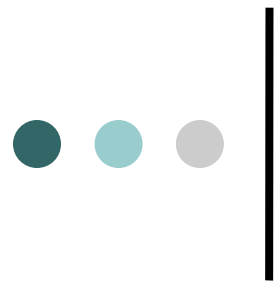


6. Uma folha contém ao menos $\lfloor (2m-1)/3 \rfloor$ chaves e não mais do que $m-1$ chaves, após as três primeiras páginas.



Árvore-B*

- As políticas de manutenção da árvore-B* devem atender às propriedades da mesma.



Próxima aula...

- Continuação das Árvores B Virtual e B+