AED2 - Algoritmos e Estr. de Dados II

Aula 10: Árvores B, B+ e B*

Prof. Aléssio Miranda Júnior
alessio@cefetmg.br

CEFET-MG - Campus Timoteo
Dep. Engenharia de Computação

Março de 2026

Sumário

O Problema do Índice

Imagine um banco com:

- 1 bilhão de registros
- Cada registro ocupa 100 bytes
- **Total $\approx$ 100 GB**

O Problema do Índice

Imagine um banco com:

- 1 bilhão de registros
- Cada registro ocupa 100 bytes
- **Total $\approx$ 100 GB**

Comparação de Velocidade

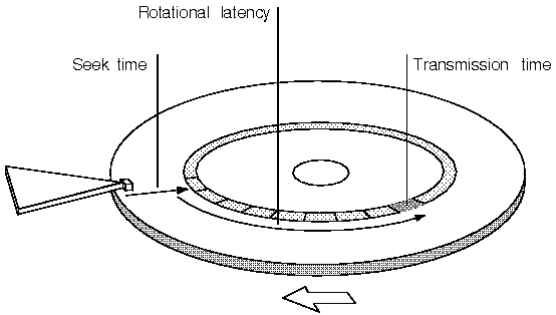
- **RAM:** 30 a 60 nanossegundos (30×10^{-9} s)
- **Disco:** 7 a 10 milissegundos (7×10^{-3} s)
- **Veredito:** A RAM é cerca de **1 milhão de vezes mais rápida**.

No entanto, o disco é mais barato, maior e persiste dados sem energia.

Custo de Acesso ao Disco

O custo é dividido em três operações:

1. **Tempo de Busca (Seek):** Mover o braço para o cilindro correto.
2. **Atraso Rotacional:** Esperar o setor girar até a cabeça.
3. **Tempo de Transferência:** Leitura real dos dados.



Fórmula da Transferência

$$\text{Tempo} = \left(\frac{\text{Bytes Transferidos}}{\text{Bytes na Trilha}} \right) \times \text{Tempo de Rotação}$$

Por que árvores binárias não resolvem?

Em uma ABB balanceada com 10^9 registros, temos altura ≈ 30.

- **Ineficiência:** Cada nó lido do disco desperdiça quase toda a página (ex: lê 4KB para pegar 16-24 bytes).
- **Custo:** 30 milissegundos por busca em disco? $30 \times 10ms = 0.3s$. Inaceitável para sistemas de alto tráfego.

O Axioma

O custo proibitivo é o **acesso ao disco**. Precisamos de uma estrutura que maximize o que ganhamos em cada acesso.

O problema principal é o **tempo de busca** (seek time) e a **latência** do hardware:

- **RAM:** 10^{-9} s.
- **Disco:** 10^{-3} s. É **1.000.000x mais lento**.

A Solução: Árvores "Largas"

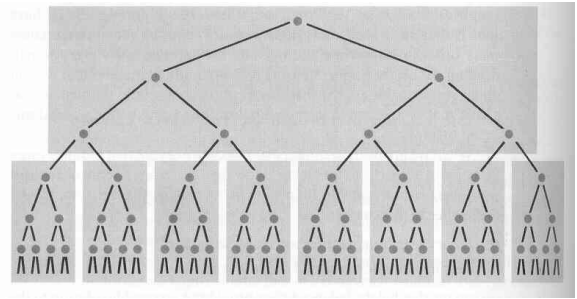
Em vez de crescer para baixo (profundidade), criamos árvores que crescem para os lados (largura), aumentando o fator de ramificação M.

$$\text{altura} \approx \log_M N$$

Modelo de Paginação

Discos trabalham com blocos de tamanho fixo (**Páginas**).

- *Insight*: Ao buscar um dado, o disco lê a página inteira (ex: 4KB) de qualquer forma.



Árvores Binárias Paginadas

Agrupamos vários nós de uma ABB dentro da mesma página de disco para reduzir o número de buscas físicas.

Vantagem da Árvore Paginada

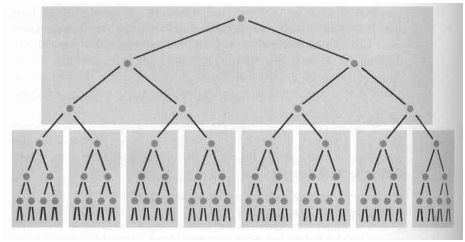


Figure: Página com 7 nós.

Exemplo: Árvore de 63 nós e página de 7.

- $(2^3 - 1) = 7$ nós por página
- $(2^6 - 1) = 63$ nós (2 seeks)
- $(2^9 - 1) = 511$ nós (3 seeks)
- $(2^{12} - 1) = 4095$ nós (4 seeks)

Eficiência no Disco

Uma busca binária em 4095 elementos exigiria **12 acessos** ao disco em uma ABB comum. Na versão paginada, apenas **4 buscas**.

Construção de Árvores Paginadas

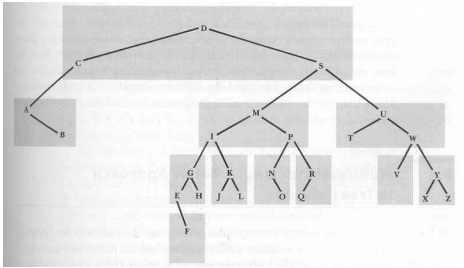


Figure: Desbalanceamento.

Como construir a árvore de forma eficiente?

- **Estático**: Se as chaves são conhecidas de antemão, basta ordenar e construir de forma balanceada.
- **Dinâmico**: Se as chaves chegam aleatoriamente, a árvore pode se desbalancear rapidamente.

Exemplo de Inserção Aleatória

`C S D T A M P I B`

O Problema da Rotação

- Em árvores não-paginadas, rotações são locais e simples.
- Em **Árvores Paginadas**, não se pode rotacionar uma página inteira facilmente.

Rearranjo Não-Local

Uma única inserção pode exigir quebras de página que se disseminam por grande parte da árvore, tornando a manutenção extremamente custosa.

Questões Não-Resolvidas

Para uma árvore em disco ser eficiente, precisamos garantir:

1. **Separadores Ideais**: Como garantir que as chaves na raiz dividam o conjunto de dados de forma equilibrada?
2. **Localidade**: Como evitar que chaves muito distantes compartilhem a mesma página?
3. **Ocupação Mínima**: Como garantir que cada página lida do disco contenha um número útil de chaves (evitar desperdício)?

A Mudança de Paradigma

A solução para esses problemas foi uma mudança radical na forma de construir a estrutura:

- Em ABBs, a árvore cresce para baixo a partir da raiz.

Solução: De Baixo para Cima

Nas Árvores B, a construção é feita de **baixo para cima**. A raiz não é fixa; ela **emerge** do conjunto de chaves conforme a árvore cresce.

Árvores B

A Solução Definitiva para Armazenamento em Disco

Artigo: R. Bayer, E. McCreight "*Organization and Maintenance of Large Ordered Indexes*". Acta-Informatica, **1**:173-189, 1972.

Organization and maintenance of large ordered indices

Full Text: Pdf

Authors: R. Bayer Mathematical and Information Sciences Laboratory
E. McCreight Mathematical and Information Sciences Laboratory

Published in:
Proceeding
SIGFIDET '70 Proceedings of the 1970 ACM SIGFIDET (now SIGMOD) Workshop on Data Description, Access and Control
Pages 107-141
ACM New York, NY, USA ©1970
table of contents doi>10.1145/1734663.1734671

1970 Article

Bibliometrics
Downloads (6 Weeks): 9
Downloads (12 Months): 47
Downloads (cumulative): 175
Citation Count: 19

Rudolf Bayer

Born	May 7, 1939 (age 75)
Nationality	German
Institutions	Technical University Munich
Alma mater	University of Illinois at Urbana–Champaign
Thesis	*Automorphism Groups and Quotients of Strongly Connected Automata and Monadic Algebras* (1966)
Doctoral advisor	Franz Edward Hohn[1]
Known for	B-tree UB-tree red-black tree
Notable awards	Cross of Merit, First class (1999), SIGMOD Edgar F. Codd Innovations Award (2001)

O nome árvores B não foi explicado:
Balanced, Broad, Boeing, Bayer.

Figure: Bayer & McCreight (1972).

2. Altura e Densidade

Considere uma Árvore B de ordem $M = 100$. Mesmo com **ocupação mínima** (50 filhos por nó):

- **Altura 1:** 50 registros ($\approx$ 1 KB de índices).
- **Altura 2:** 2.500 registros.
- **Altura 3:** 125.000 registros.
- **Altura 4:** 6.250.000 registros.
- **Altura 5:** 312.500.000 registros.

Eficiência de I/O

Com apenas **5 leituras de disco**, conseguimos indexar centenas de milhões de registros. Em uma árvore binária, seriam necessários aproximadamente **28 acessos** para o mesmo volume.

B-Tree: vários filhos por nó (ordem M). Nó $\leq M-1$ chaves; folhas no mesmo nível; chaves ordenadas.

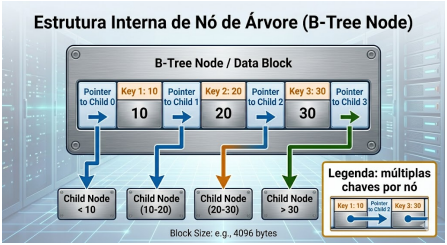


Figure: Nó B-Tree.

Propriedades da Árvore B de ordem M

1. **Raiz:** tem pelo menos 2 filhos se não for folha.
2. **Nós internos:** têm entre $\lceil M/2 \rceil$ e M filhos.
3. **Chaves:** um nó com k filhos tem $k-1$ chaves.
4. **Equilíbrio:** Todas as folhas estão no mesmo nível.

Ordem da Árvore B

Se a árvore possui ordem M:

- máximo de filhos $= M$
- máximo de chaves $= M - 1$
- mínimo de filhos $= \lceil M/2 \rceil$

Ordem da Árvore B

Se a árvore possui ordem M:

- máximo de filhos $= M$
- máximo de chaves $= M - 1$
- mínimo de filhos $= \lceil M/2 \rceil$

Consequência

A árvore permanece sempre balanceada.

Altura de uma Árvore B

Número máximo de registros:

$$N \approx M^h$$

Altura de uma Árvore B

Número máximo de registros:

$$N \approx M^h$$

Exemplo:

- $M = 100$
- $h = 3$

Altura de uma Árvore B

Número máximo de registros:

$$N \approx M^h$$

Exemplo:

- $M = 100$
- $h = 3$

$$100^3 = 1.000.000$$

Altura de uma Árvore B

Número máximo de registros:

$$N \approx M^h$$

Exemplo:

- $M = 100$
- $h = 3$

$$100^3 = 1.000.000$$

Apenas 3 acessos ao disco

Busca em Árvore B

Processo:

1. buscar chave dentro do nó
2. escolher filho correto
3. repetir até folha

Busca em Árvore B

Processo:

1. buscar chave dentro do nó
2. escolher filho correto
3. repetir até folha

Complexidade:

$$O(\log_M N)$$

4. Inserção: Árvore B (Página=4)

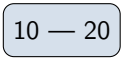
Observe que o Split ocorre apenas quando a página excede 4 chaves.

Sequência: **10**, 20, 30, 40, 50, 60

4. Inserção: Árvore B (Página=4)

Observe que o Split ocorre apenas quando a página excede 4 chaves.

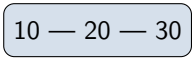
Sequência: 10, **20**, 30, 40, 50, 60



4. Inserção: Árvore B (Página=4)

Observe que o Split ocorre apenas quando a página excede 4 chaves.

Sequência: 10, 20, **30**, 40, 50, 60



4. Inserção: Árvore B (Página=4)

Observe que o Split ocorre apenas quando a página excede 4 chaves.

Sequência: 10, 20, 30, **40**, 50, 60

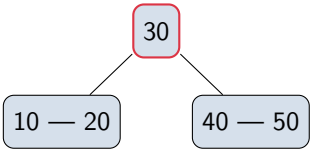
10 — 20 — 30 — 40

4. Inserção: Árvore B (Página=4)

Observe que o Split ocorre apenas quando a página excede 4 chaves.

Sequência: 10, 20, 30, 40, **50**, 60

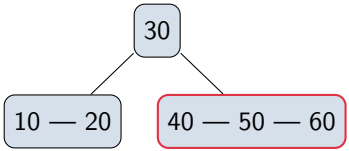
Inserção de 50 causou overflow! Promove o 30.



4. Inserção: Árvore B (Página=4)

Observe que o Split ocorre apenas quando a página excede 4 chaves.

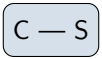
Sequência: 10, 20, 30, 40, 50, 60



B-Tree (Standard) - Passo 1

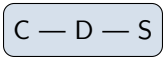
Sequência: **C** S D T A M P I B W N G U ...

B-Tree (Standard) - Passo 2



Sequência: C **S** D T A M P I B W N G U ...

B-Tree (Standard) - Passo 3



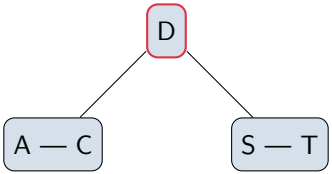
Sequência: C S **D** T A M P I B W N G U ...

B-Tree (Standard) - Passo 4

C — D — S — T

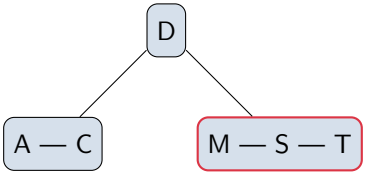
Sequência: C S D **T** A M P I B W N G U ...

B-Tree (Standard) - Passo 5



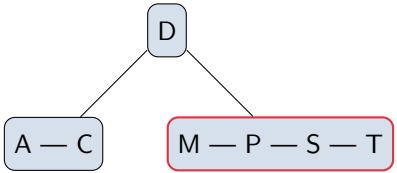
Sequência: C S D T **A** M P I B W N G U ...

B-Tree (Standard) - Passo 6



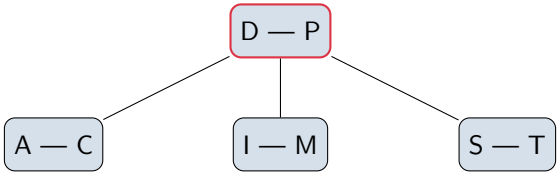
Sequência: C S D T A **M** P I B W N G U ...

B-Tree (Standard) - Passo 7



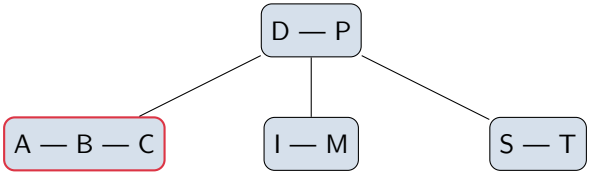
Sequência: C S D T A M **P** I B W N G U ...

B-Tree (Standard) - Passo 8



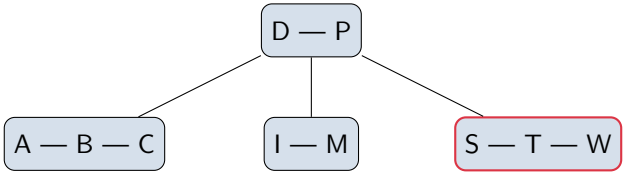
Sequência: C S D T A M P I B W N G U ...

B-Tree (Standard) - Passo 9



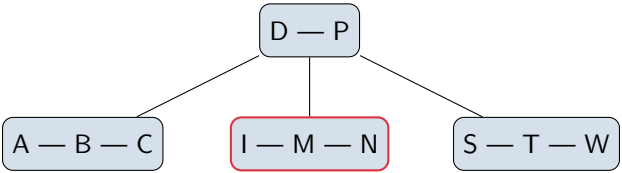
Sequência: C S D T A M P I **B** W N G U ...

B-Tree (Standard) - Passo 10



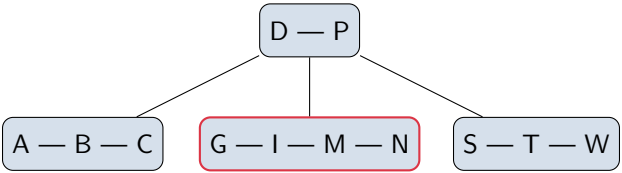
Sequência: C S D T A M P I B **W** N G U ...

B-Tree (Standard) - Passo 11



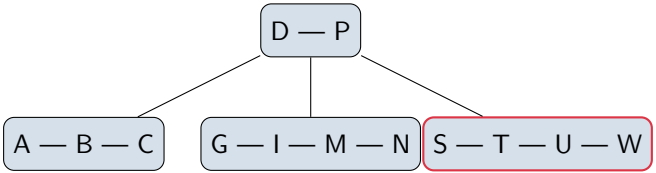
Sequência: C S D T A M P I B W **N** G U ...

B-Tree (Standard) - Passo 12



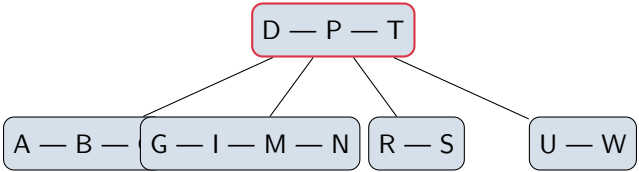
Sequência: C S D T A M P I B W N **G** U ...

B-Tree (Standard) - Passo 13



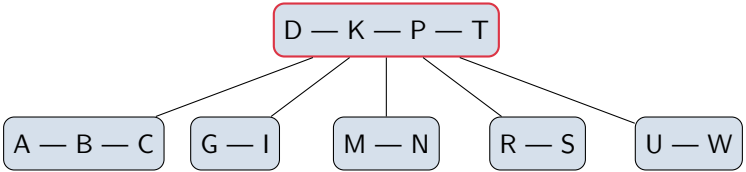
Sequência: C S D T A M P I B W N G **U** ...

B-Tree (Standard) - Passo 14



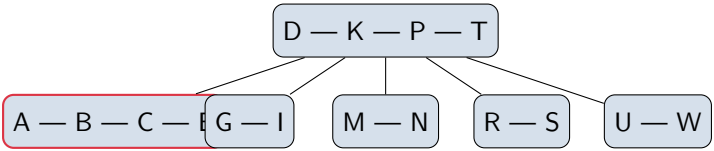
Sequência: C... R K E H O L J Y Q Z F X V

B-Tree (Standard) - Passo 15



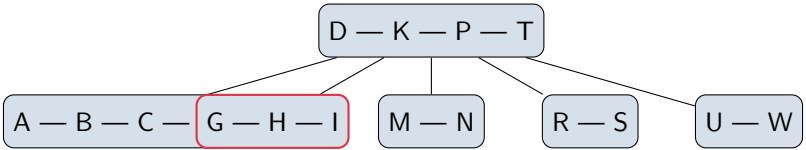
Sequência: C... R **K** E H O L J Y Q Z F X V

B-Tree (Standard) - Passo 16



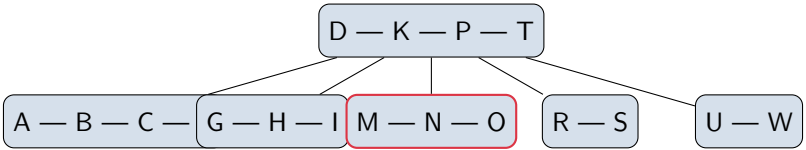
Sequência: C... R K **E** H O L J Y Q Z F X V

B-Tree (Standard) - Passo 17



Sequência: C... R K E **H** O L J Y Q Z F X V

B-Tree (Standard) - Passo 18



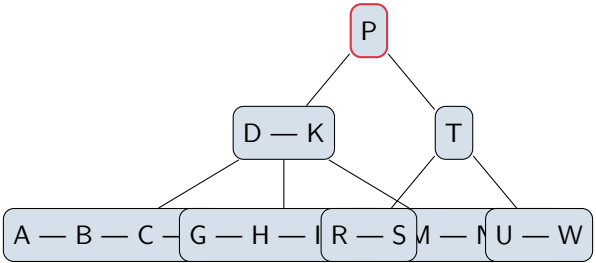
Sequência: C... R K E H O L J Y Q Z F X V

B-Tree (Standard) - Passo 19

...

Sequência: C... R K E H O **L** J Y Q Z F X V

B-Tree (Standard) - Passo 20



Sequência: C... R K E H O L J Y Q Z F X V

B-Tree (Standard) - Passo 21

...

Sequência: C... R K E H O L J Y Q Z F X V

B-Tree (Standard) - Passo 22

...

Sequência: C... R K E H O L J Y Q Z F X V

B-Tree (Standard) - Passo 23

...

Sequência: C... R K E H O L J Y Q Z F X V

B-Tree (Standard) - Passo 24

...

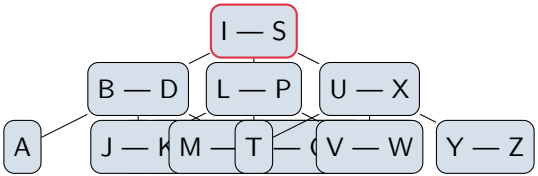
Sequência: C... R K E H O L J Y Q Z **F** X V

B-Tree (Standard) - Passo 25

...

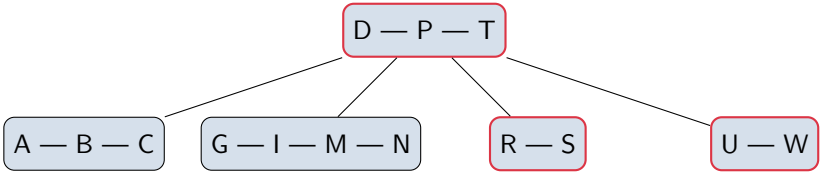
Sequência: C... R K E H O L J Y Q Z F **X** V

B-Tree (Standard) - Passo 26



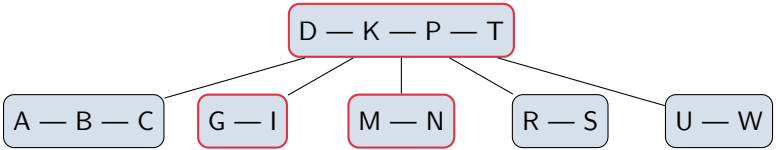
Sequência: C... R K E H O L J Y Q Z F X **V**

Animação B-Tree (Página=4) - Etapa 09



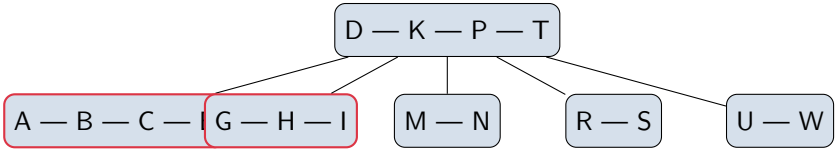
Entrada: C...U, **R**, K...

Animação B-Tree (Página=4) - Etapa 10



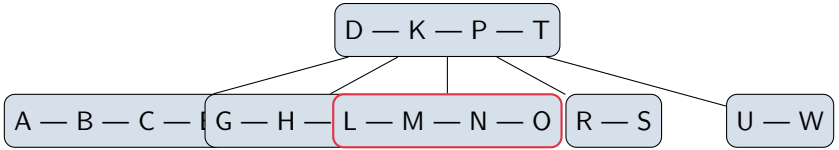
Entrada: C...R, **K**, E, H...

Animação B-Tree (Página=4) - Etapa 11



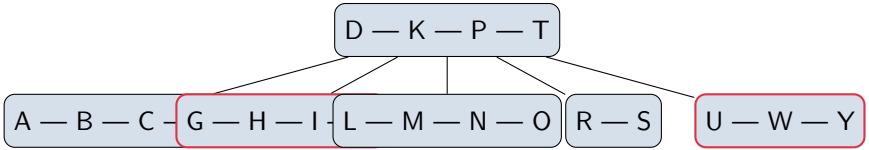
Entrada: C...K, **E, H**, O, L...

Animação B-Tree (Página=4) - Etapa 12



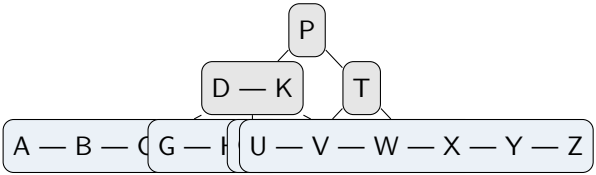
Entrada: C...H, **O, L**, J, Y...

Animação B-Tree (Página=4) - Etapa 13



Entrada: C...L, **J, Y**, Q...

Animação B-Tree (Página=4) - Etapa 14



Estado Final (B-Tree)

Árvore com chaves distribuídas por todos os níveis. Diferente da B+, quando o dado é encontrado em um nó de índice, a busca termina ali.

5. Árvore B+ (B-Plus Tree)

É a variação padrão para Bancos de Dados Relacionais e Sistemas de Arquivos.

Diferenças Cruciais

- **Nós Internos = Apenas Índices:** Não guardam os registros, apenas cópias das chaves para guiar a busca.
- **Folhas = Dados + Corrente:** Todas as chaves reais estão nas folhas, que são ligadas entre si.

Por que usar B+?

Permite **Varreduras Sequenciais** percorrrendo apenas a lista ligada das folhas.

6. Inserção: Árvore B+ (M=4)

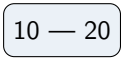
O valor promovido permanece na folha.

Sequência: 10, 20, 30, ...

6. Inserção: Árvore B+ (M=4)

O valor promovido permanece na folha.

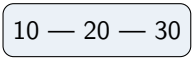
Sequência: 10, 20, 30, ...



6. Inserção: Árvore B+ (M=4)

O valor promovido permanece na folha.

Sequência: 10, 20, 30, ...



6. Inserção: Árvore B+ (M=4)

O valor promovido permanece na folha.

Sequência: 10, 20, 30, **40**, ...

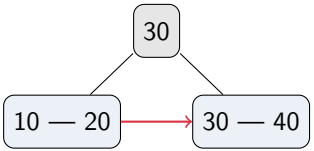
30 sobe e fica na folha!

10 — 20 — 30 — 40

6. Inserção: Árvore B+ (M=4)

O valor promovido permanece na folha.

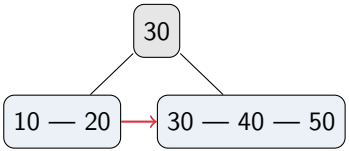
Sequência: 10, 20, 30, 40, **50**, ...



6. Inserção: Árvore B+ (M=4)

O valor promovido permanece na folha.

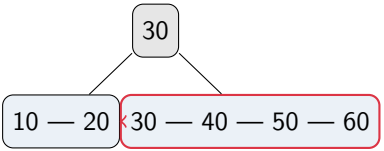
Sequência: 10, 20, 30, 40, **50**, ...



6. Inserção: Árvore B+ (M=4)

O valor promovido permanece na folha.

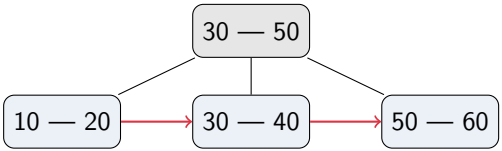
Sequência: 10, 20, 30, 40, 50, **60**



6. Inserção: Árvore B+ (M=4)

O valor promovido permanece na folha.

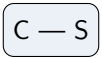
Sequência: 10, 20, 30, 40, 50, 60



B+ Tree (Lógica do Livro) - Passo 1

Sequência: **C** S D T A M P I B W N G U ...

B+ Tree (Lógica do Livro) - Passo 2



Sequência: C **S** D T A M P I B W N G U ...

B+ Tree (Lógica do Livro) - Passo 3

C — D — S

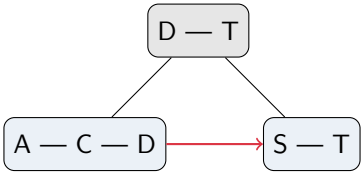
Sequência: C S D T A M P I B W N G U ...

B+ Tree (Lógica do Livro) - Passo 4

C — D — S — T

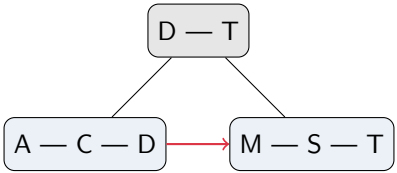
Sequência: C S D **T** A M P I B W N G U ...

B+ Tree (Lógica do Livro) - Passo 5



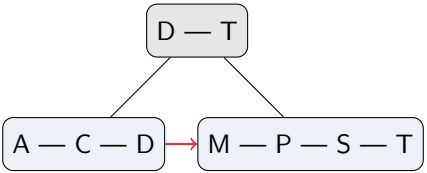
Sequência: C S D T **A** M P I B W N G U ...

B+ Tree (Lógica do Livro) - Passo 6



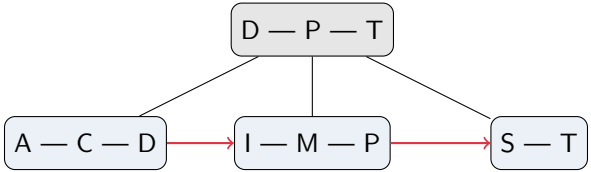
Sequência: C S D T A **M** P I B W N G U ...

B+ Tree (Lógica do Livro) - Passo 7



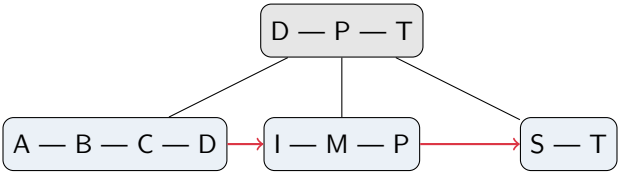
Sequência: C S D T A M **P** I B W N G U ...

B+ Tree (Lógica do Livro) - Passo 8



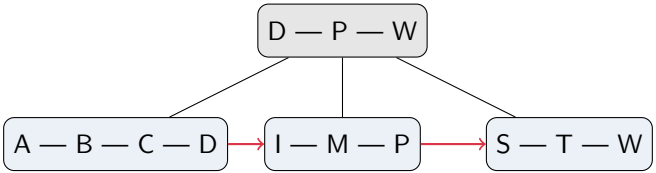
Sequência: C S D T A M P I B W N G U ...

B+ Tree (Lógica do Livro) - Passo 9



Sequência: C S D T A M P I **B** W N G U ...

B+ Tree (Lógica do Livro) - Passo 10



Sequência: C S D T A M P I B **W** N G U ...

B+ Tree (Lógica do Livro) - Passo 11

...

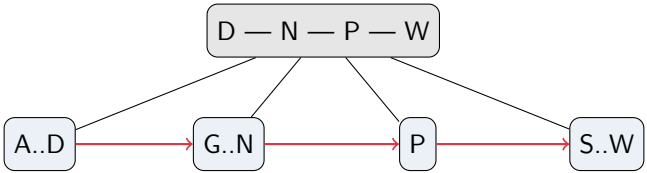
Sequência: C S D T A M P I B W **N** G U ...

B+ Tree (Lógica do Livro) - Passo 12

...

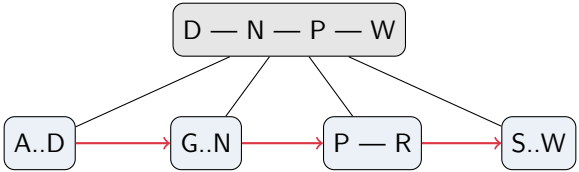
Sequência: C S D T A M P I B W N G U ...

B+ Tree (Lógica do Livro) - Passo 13



Sequência: C S D T A M P I B W N G **U** ...

B+ Tree (Lógica do Livro) - Passo 14



Sequência: C... R K E H O L J Y Q Z F X V

B+ Tree (Lógica do Livro) - Passo 15

...

Sequência: C... R K E H O L J Y Q Z F X V

B+ Tree (Lógica do Livro) - Passo 16

...

Sequência: C... R K **E** H O L J Y Q Z F X V

B+ Tree (Lógica do Livro) - Passo 17

...

Sequência: C... R K E **H** O L J Y Q Z F X V

B+ Tree (Lógica do Livro) - Passo 18

...

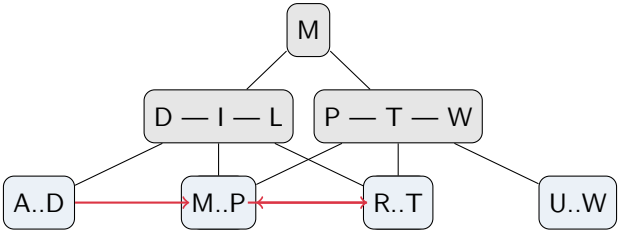
Sequência: C... R K E H O L J Y Q Z F X V

B+ Tree (Lógica do Livro) - Passo 19

...

Sequência: C... R K E H O **L** J Y Q Z F X V

B+ Tree (Lógica do Livro) - Passo 20



Sequência: C... R K E H O L J Y Q Z F X V

B+ Tree (Lógica do Livro) - Passo 21

...

Sequência: C... R K E H O L J **Y** Q Z F X V

B+ Tree (Lógica do Livro) - Passo 22

...

Sequência: C... R K E H O L J Y Q Z F X V

B+ Tree (Lógica do Livro) - Passo 23

...

Sequência: C... R K E H O L J Y Q **Z** F X V

B+ Tree (Lógica do Livro) - Passo 24

...

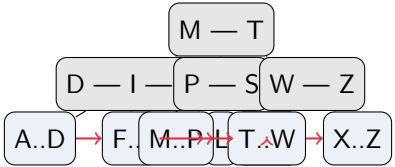
Sequência: C... R K E H O L J Y Q Z **F** X V

B+ Tree (Lógica do Livro) - Passo 25

...

Sequência: C... R K E H O L J Y Q Z F **X** V

B+ Tree (Lógica do Livro) - Passo 26



Sequência: C... R K E H O L J Y Q Z F X **V**

7. Árvore B* (B-Star Tree)

É a alternativa para quem busca o máximo de eficiência de espaço.

Redistribuição vs Split

Se o nó estiver cheio, a B* tenta **compartilhar** com o irmão. O split só ocorre se o vizinho também estiver cheio.

Resultado

Garante ocupação mínima de **2/3** (66%). Árvores mais densas e ainda mais baixas.

Motivação da B* Tree

B-trees permitem nós com apenas:

50% ocupação

Motivação da B* Tree

B-trees permitem nós com apenas:

50% ocupação

Isso desperdiça espaço em disco.

Motivação da B* Tree

B-trees permitem nós com apenas:

$$50\% \text{ ocupação}$$

Isso desperdiça espaço em disco.
B* exige:

$$\frac{2}{3} \text{ ocupação mínima}$$

Split em B*

Quando dois nós estão cheios:

Split em B*

Quando dois nós estão cheios:

- redistribui chaves
- cria três nós

Split em B*

Quando dois nós estão cheios:

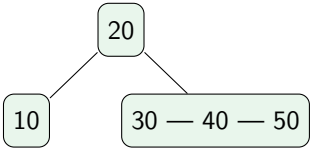
- redistribui chaves
- cria três nós

Resultado:

árvore mais densa

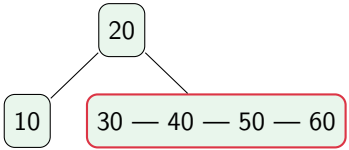
8. Inserção: Árvore B* (M=4)

Simulando a rotação para evitar o split.



8. Inserção: Árvore B* (M=4)

Simulando a rotação para evitar o split.

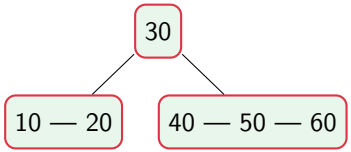


Overflow! Vizinho esquerdo tem espaço.

8. Inserção: Árvore B* (M=4)

Simulando a rotação para evitar o split.

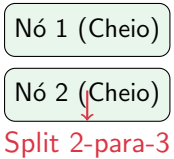
Rotação concluída: 20 desceu e 30 subiu.



Árvore B*: Maximizando a Ocupação

A Árvore B* é uma variante que busca manter os nós ainda mais cheios:

- **Ocupação Mínima:** 2/3 (em vez de 1/2).
- **Split Adiado:** Quando um nó enche, ele tenta "vazar" chaves para o vizinho antes de dividir.
- **Divisão 2-em-3:** Quando o split é inevitável, dois nós cheios são transformados em três nós 2/3 cheios.



9. O Desafio da Remoção

A remoção deve manter a ocupação mínima de $\lceil M/2 \rceil$.

Underflow Menos chaves que o mínimo.

Redistribuição "Pede emprestado" de um irmão.

Concatenação Funde (merge) o nó com um irmão.

10. B-Tree vs. Hash Index

Operação	B+ Tree	Hash Index
Busca Exata	$O(\log_M N)$	$O(1)$
Busca Faixa	Excelente	Não suporta
Ordenação	Nativa	Requer Sort extra

Conclusão: Árvores B são ideais para bancos de dados.

Resumo Final das Variantes

Tipo	Ocupação	Split	Ponteiros de Folha
B	50%	1 node $\rightarrow$ 2	Não
B+	50%	1 node $\rightarrow$ 2 (com cópia)	Sim (Sequencial)
B*	66%	2 nodes $\rightarrow$ 3	Não

Comparação Final

Estrutura	ocupação	folhas ligadas	uso
B-tree	50%	não	teoria
B+ tree	50%	sim	bancos
B* tree	66%	não	sistemas de arquivos

11. Mão na Massa

Dada uma **Árvore B de Ordem 3** (Máx 2 chaves), desenhe o estado final:

1. Inserir: 5, 10, 15, 20, 25
2. Remover: 5

Estudo de Caso: Inserção Passo a Passo

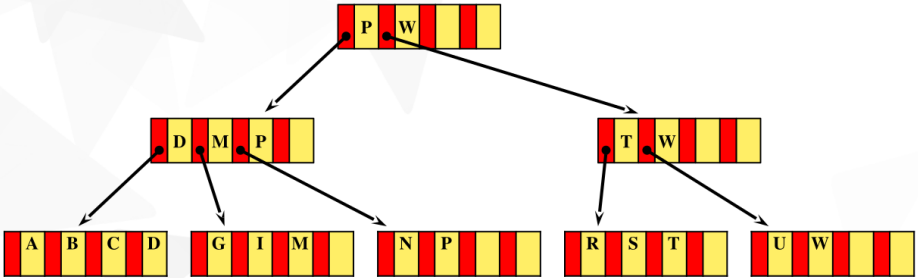
Parâmetros do Exemplo

- **Sequência:** 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
- **Ordem**: $M = 4$ (Máximo de 3 chaves por nó)
- **Crescimento**: A árvore final terá altura 3 para as 26 chaves.

Regra de Divisão

O número de nós afetados por qualquer inserção nunca é maior do que dois nós por nível (um mudado e outro criado por uma divisão).

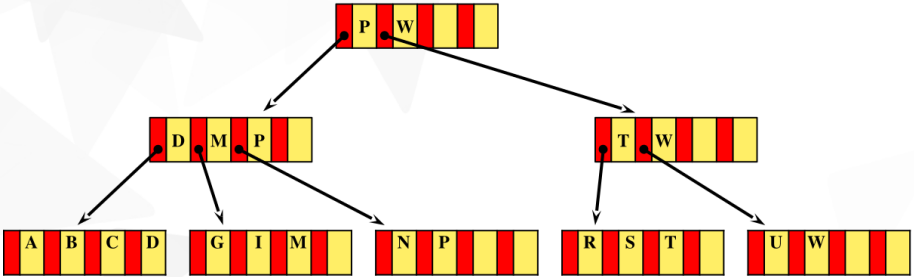
Inserção B-Tree (M=4) - Etapa 01



Sequência: C S

D T AM P I B W N G U R K E H O L J Y Q Z F X V

Inserção B-Tree (M=4) - Etapa 02



Sequência: C S

D T AM P I B W N G U R K E H O L J Y Q Z F X V

Inserção B-Tree (M=4) - Etapa 04

- Input Sequence:

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*

Insertion of C, S, D, T
into the initial page

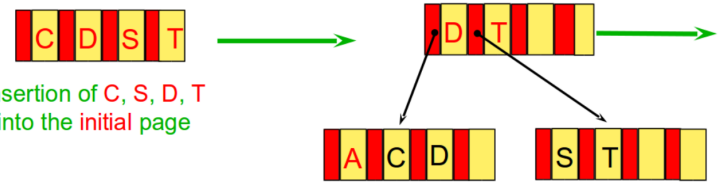
Sequência: C S

D T AM P I B W N G U R K E H O L J Y Q Z F X V

Inserção B-Tree (M=4) - Etapa 05

- Input Sequence:

  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



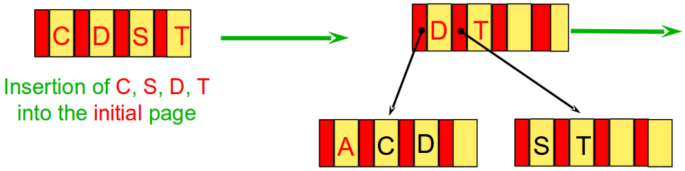
Sequência: C S D T A M P I B W

N G U R K E H O L J Y Q Z F X V

Inserção B-Tree (M=4) - Etapa 06

- Input Sequence:

  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*



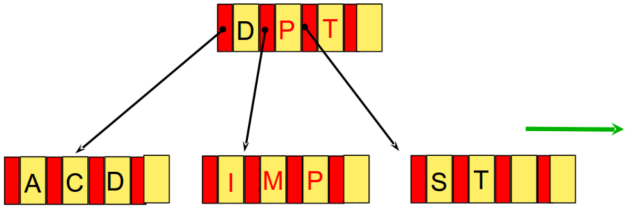
Sequência: C S D T AM P I B W N G U

R K E H O L J Y Q Z F X V

Inserção B-Tree (M=4) - Etapa 07

❍ Input Sequence:

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



M and **P** are inserted into the rightmost leaf node,
then insertion of **I** causes it to split

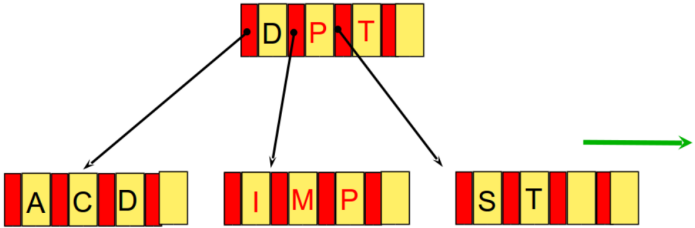
Sequência: C S D T AM P I B W N G U

R K E H O L J Y Q Z F X V

Inserção B-Tree (M=4) - Etapa 08

- Input Sequence:

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



M and **P** are inserted into the rightmost leaf node,
then insertion of **I** causes it to split

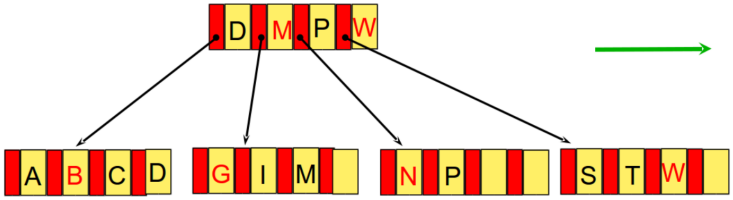
Sequência: C S D T AM P I B

W N G U R K E H O L J Y Q Z F X V

Inserção B-Tree (M=4) - Etapa 09

❍ Input Sequence:

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



Insertions of B,W,N, and G into leaf nodes causes another split and the root is now full

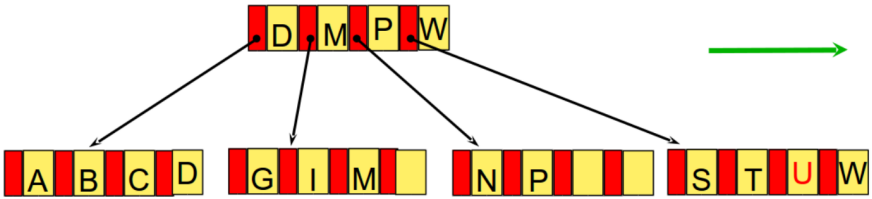
Sequência: C S D T AM P I B W N G

U R K E H O L J Y Q Z F X V

Inserção B-Tree (M=4) - Etapa 10

❍ Input Sequence:

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



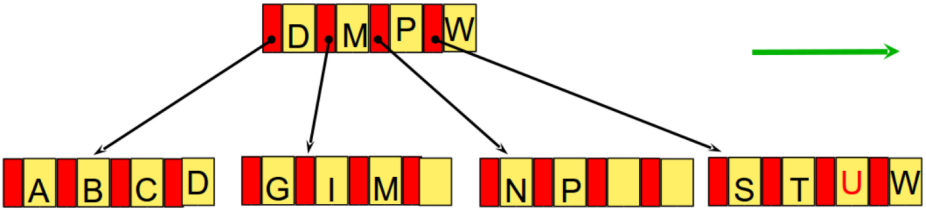
Insertion of U proceeds without incident, but R would have to be inserted into the rightmost leaf, which is full

Sequência: C S D T

AM P I B W N G U R K E H O L J Y Q Z F X V

- Input Sequence:

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



Insertion of U proceeds without incident, but R would have to be inserted into the rightmost leaf, which is full

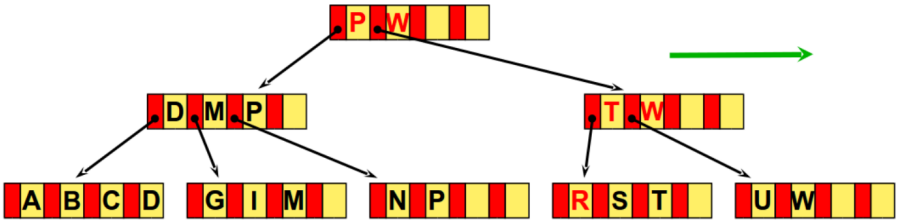
Sequência: C S D

T AM P I B W N G U R K E H O L J Y Q Z F X V

Inserção B-Tree (M=4) - Etapa 12

- Input Sequence:

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



Insertion of R causes the rightmost leaf node to split, insertion into the root causes the **root to split** and the tree grows to level three

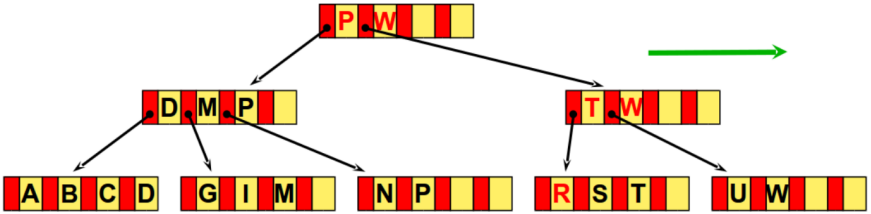
Sequência: C S D T

AM P I B W N G U R K E H O L J Y Q Z F X V

Inserção B-Tree (M=4) - Etapa 13

❍ Input Sequence:

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



Insertion of R causes the rightmost leaf node to split, insertion into the root causes the **root to split** and the tree grows to level three

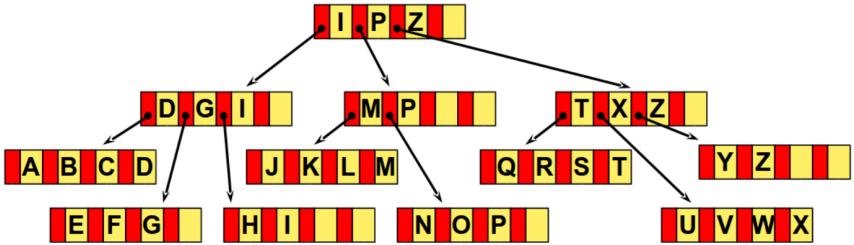
Sequência: C S D T AM

P I B W N G U R K E H O L J Y Q Z F X V

Inserção B-Tree (M=4) - Etapa 14

- Input Sequence:

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**



Insertions of F, X, and V finish the insertion of the alphabet

Sequência: C S D T AM P I

B W N G U R K E H O L J Y Q Z F X V