

# Árvores-B: Definição e Complexidade

Prof. Ms. Anderson Canale Garcia

Adaptado de:

Profa. Dra. Cristina D. Aguiar

---

# Definição Formal (1/3)

- Árvore-B com ordem  $m$ 
    - cada página (ou nó) deve ser da seguinte forma  
 $\langle \langle C_1, PR_1 \rangle, \langle C_2, PR_2 \rangle, \dots, \langle C_{q-1}, PR_{q-1} \rangle, P_1, P_2, \dots, P_{q-1}, P_q \rangle$ , onde  $(q \leq m)$ 
      - cada  $C_i$  ( $1 \leq i \leq q - 1$ ) é uma chave de busca
      - cada  $P_{Ri}$  ( $1 \leq i \leq q - 1$ ) é um campo de referência para o registro no arquivo de dados que contém o registro de dados correspondente a  $C_i$
      - cada  $P_j$  ( $1 \leq j \leq q$ ) é um campo de referência para uma subárvore ou assume o valor -1 caso não exista subárvore (ou seja, caso seja um nó folha)
-

# Definição Formal (2/3)

- Árvore-B com ordem  $m$ 
    - dentro de cada página,  $C_1 < C_2 < \dots < C_{q-1}$
    - para todos os valores  $X$  da chave na subárvore apontada por  $P_i$ 
      - $C_{i-1} < X < C_i$  para  $1 < i < q$
      - $X < C_i$  para  $i = 1$
      - $C_{i-1} < X$  para  $i = q$
-

# Definição Formal (3/3)

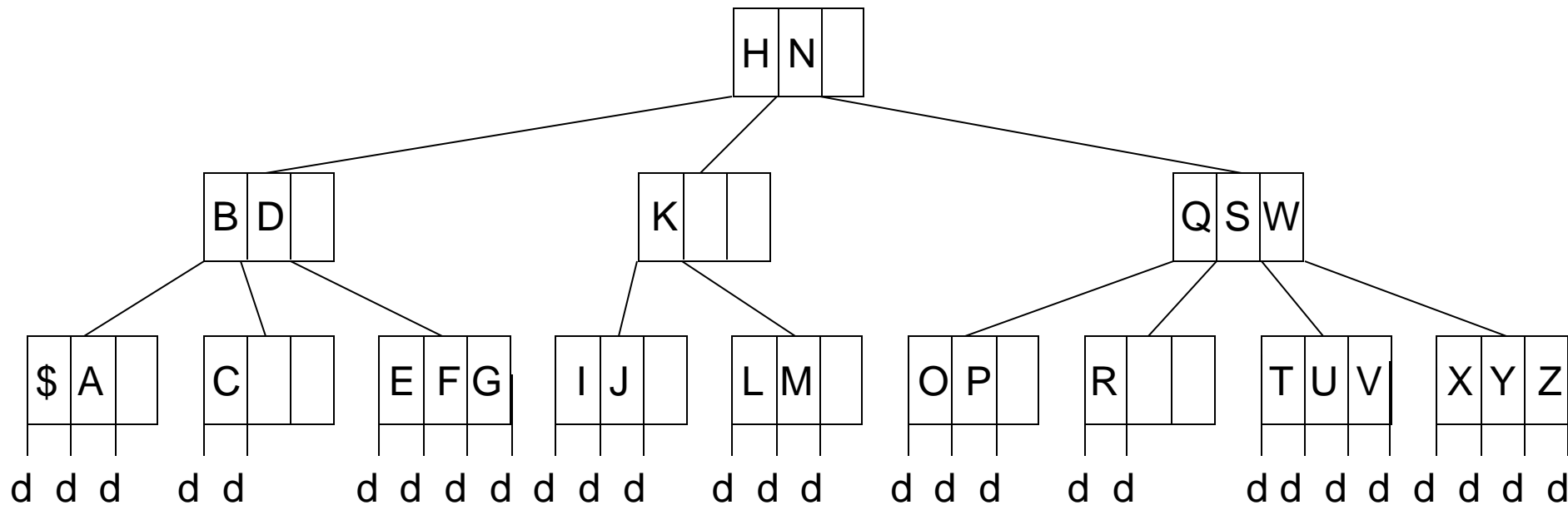
- Árvore-B com ordem  $m$ 
    - cada página possui um máximo de  $m$  descendentes
    - cada página, exceto a raiz e as folhas, possui no mínimo  $\lceil m/2 \rceil$  descendentes → taxa de ocupação
    - a raiz possui pelo menos 2 descendentes, a menos que seja um nó folha
    - todas as folhas aparecem no mesmo nível
    - uma página interna com  $k$  descendentes contém  $k-1$  chaves
    - uma folha possui no mínimo  $\lceil m/2 \rceil - 1$  chaves e no máximo  $m - 1$  chaves → taxa de ocupação
-

# Complexidade (Pior Caso)

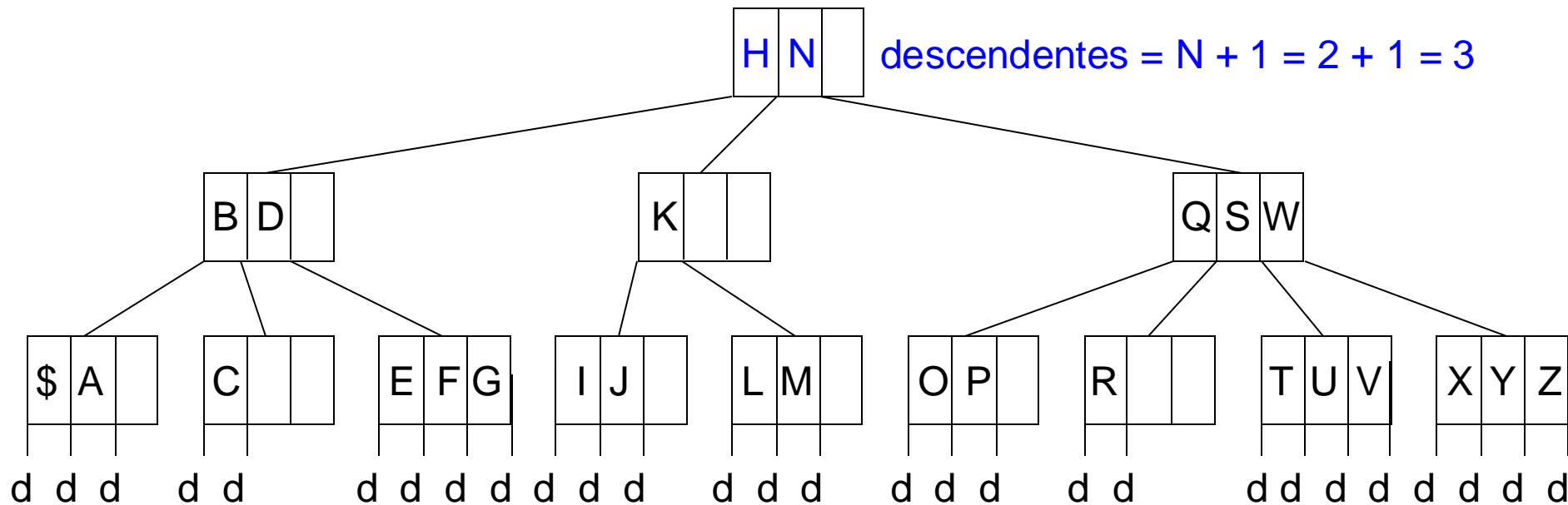
- Profundidade do caminho de busca
  - número máximo de acessos a disco

# Característica

número de descendentes de um nível da árvore-B = número de chaves contidas no nível em questão e em todos os níveis acima (N) + 1

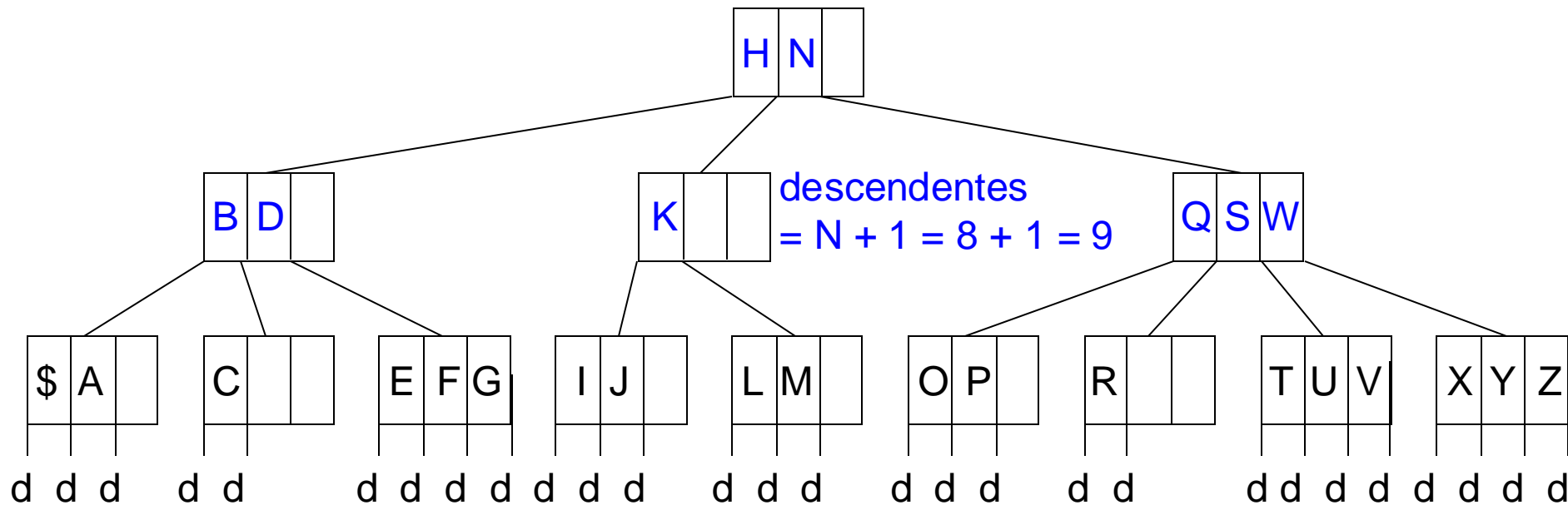


# Característica

$$\begin{aligned} \text{número de descendentes} &= \text{número de chaves contidas} + 1 \\ \text{de um nível da árvore-B} &\quad \text{no nível em questão e} \\ &\quad \text{em todos os níveis acima (N)} \end{aligned}$$


# Característica

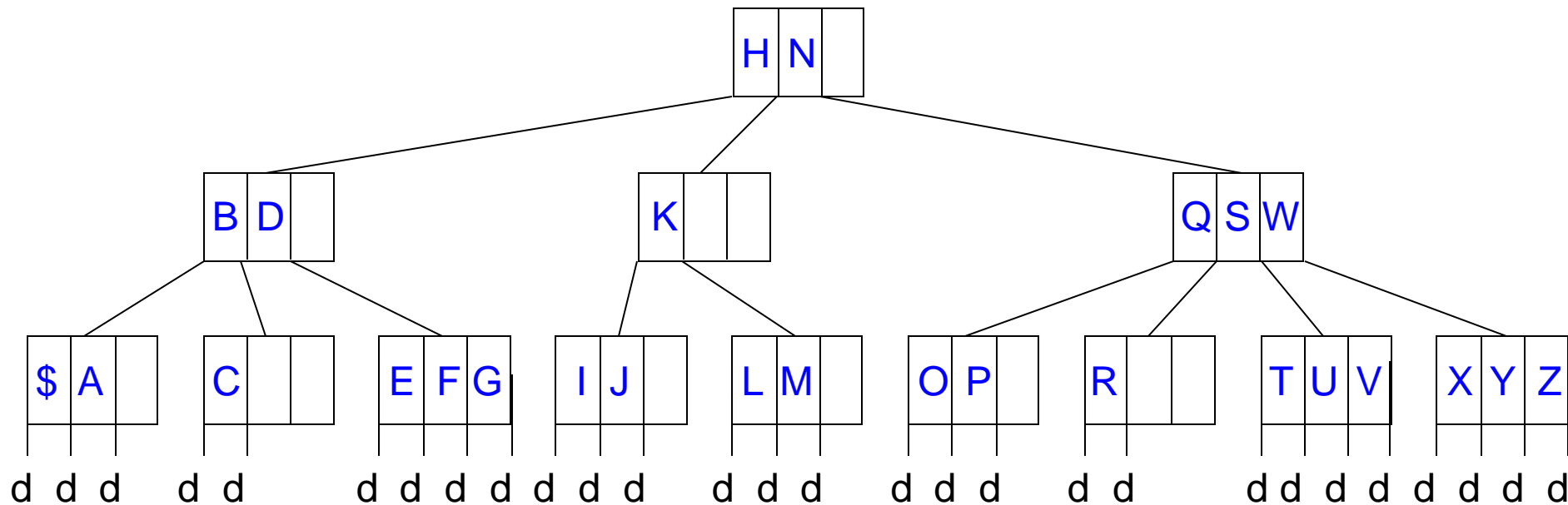
número de descendentes de um nível da árvore-B = número de chaves contidas no nível em questão e em todos os níveis acima (N) + 1





# Característica

número de descendentes de um nível da árvore-B = número de chaves contidas no nível em questão e em todos os níveis acima (N) + 1



$$\text{descendentes} = N + 1 = 27 + 1 = 28$$

# Característica

- Propriedades da árvore-B de ordem  $m$ 
  - cálculo do número mínimo de descendentes de um nível (**análise do pior caso**)

| nível | número mínimo de descendentes   |
|-------|---|
| 1     | 2   |
| 2     | $2 \times \lceil m/2 \rceil$  |
| 3     | $2 \times \lceil m/2 \rceil \times \lceil m/2 \rceil = 2 \times \lceil m/2 \rceil^2$                          |
| 4     | $2 \times \lceil m/2 \rceil \times \lceil m/2 \rceil \times \lceil m/2 \rceil = 2 \times \lceil m/2 \rceil^3$ |
| ...   | ...   |
| d     | $2 \times \lceil m/2 \rceil^{d-1}$  |

nível 1 (raiz):  
tem no mínimo  
2  
descendentes

# Observação 2

- Propriedades da árvore-B de ordem  $m$ 
  - cálculo do número mínimo de descendentes de um nível (*análise do pior caso*)

| nível | número mínimo de descendentes   |
|-------|---|
| 1     | 2   |
| 2     | $2 \times \lceil m/2 \rceil$  |
| 3     | $2 \times \lceil m/2 \rceil \times \lceil m/2 \rceil = 2 \times \lceil m/2 \rceil^2$                          |
| 4     | $2 \times \lceil m/2 \rceil \times \lceil m/2 \rceil \times \lceil m/2 \rceil = 2 \times \lceil m/2 \rceil^3$ |
| ...   | ...   |
| d     | $2 \times \lceil m/2 \rceil^{d-1}$  |

nível 2: tem 2 páginas a partir da raiz, sendo cada uma com no mínimo  $\lceil m/2 \rceil$  descendentes

# Observação 2

- Propriedades da árvore-B de ordem  $m$ 
  - cálculo do número mínimo de descendentes de um nível (*análise do pior caso*)

| nível | número mínimo de descendentes   |
|-------|---|
| 1     | 2   |
| 2     | $2 \times \lceil m/2 \rceil$  |
| 3     | $2 \times \lceil m/2 \rceil \times \lceil m/2 \rceil = 2 \times \lceil m/2 \rceil^2$                          |
| 4     | $2 \times \lceil m/2 \rceil \times \lceil m/2 \rceil \times \lceil m/2 \rceil = 2 \times \lceil m/2 \rceil^3$ |
| ...   | ...   |
| d     | $2 \times \lceil m/2 \rceil^{d-1}$  |

nível 3: cada uma das 2 páginas a partir da raiz tem  $\lceil m/2 \rceil$  páginas, sendo cada uma com no mínimo  $\lceil m/2 \rceil$  descendentes

# Observação 2

- Propriedades da árvore-B de ordem  $m$ 
  - cálculo do número mínimo de descendentes de um nível (*análise do pior caso*)

| nível | número mínimo de descendentes   |
|-------|---|
| 1     | 2   |
| 2     | $2 \times \lceil m/2 \rceil$  |
| 3     | $2 \times \lceil m/2 \rceil \times \lceil m/2 \rceil = 2 \times \lceil m/2 \rceil^2$                          |
| 4     | $2 \times \lceil m/2 \rceil \times \lceil m/2 \rceil \times \lceil m/2 \rceil = 2 \times \lceil m/2 \rceil^3$ |
| ...   | ...   |
| $d$   | $2 \times \lceil m/2 \rceil^{d-1}$  |

para qualquer nível  $d$ , com exceção da raiz (nível 1)



# Complexidade (Pior Caso)

- Número de chaves (N)
    - $N + 1$  descendentes no nível das folhas
  - Profundidade da árvore-B no nível das folhas
    - $d$
  - Relacionamento
    - $N + 1$  descendentes e
    - número mínimo de descendentes da árvore-B com profundidade  $d$
-

# Exemplo

- Tamanho da página de disco
    - árvore-B de ordem 512
      - 511 chaves/página
    - número de chaves
      - 1.000.000 de chaves
    - ⇒ número de níveis que pode ser atingido?
-

# Exemplo

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

–  $m = 512$

–  $N = 1.000.000$

–  $d \leq 1 + \log_{256} (500.000,50) \Rightarrow d \leq 3,37$

– acesso a disco adicional: arquivo de dados

a árvore possui não mais  
do que 3 níveis de altura





# Complexidade

- O número de acessos ao disco exigidos para a maioria das operações em uma árvore-B é proporcional a sua altura (Cormen, T.)
  - Uma árvore-B com ***n*** itens tem complexidade de E/S
    - $O(\log_B n)$  para operações de pesquisa/atualização
    - Usa  $(n/B)$  blocos
  - **B** é o tamanho de um bloco
-

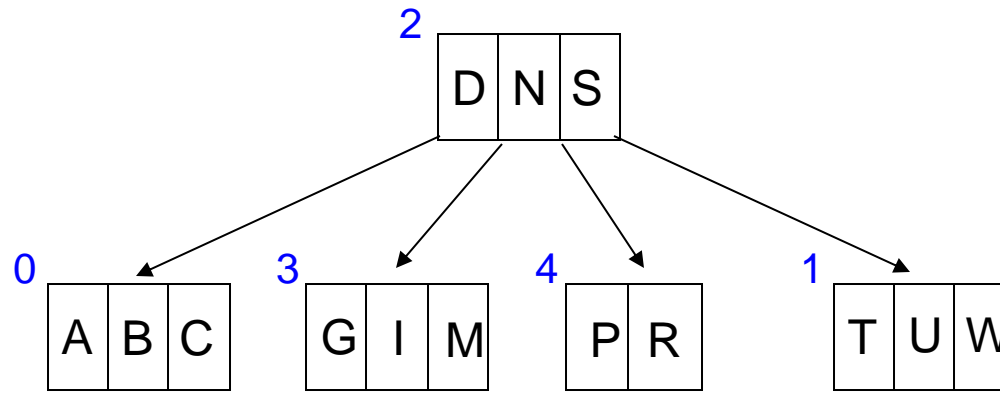
# IMPLEMENTAÇÃO

---

# Páginas da Árvore-B

- Estrutura de dados
    - determina cada página de disco
    - pode ser implementada de diferentes formas
  - Implementação adotada
    - contador de ocupação  $\Rightarrow$  número de chaves por página
    - chaves  $\Rightarrow$  chaves de busca
    - ponteiros  $\Rightarrow$  referências para os registros do arquivo de índice  $\Rightarrow$  RRNs para páginas
-

# Arquivo da Árvore-B



|                 | contador<br>de ocupação | chaves e RRNs<br>para arquivo de dados | referências para os filhos<br>no arquivo de índice<br>ou -1 (sem filhos) |
|-----------------|-------------------------|--|--|
| <i>página 2</i> | 3                       | D,1   N,7   S,9                        | 0   3   4   1  |
| <i>página 4</i> | 2                       | P,3   R,4                              | -1   -1   -1   -1  |

# Declaração em C - Estrutura

```
#define ORDEM 5

typedef struct {
    int chave;
    int RRN;
} CHAVE;

typedef struct pagina {
    short ocupacao;           // Taxa de ocupação atual da página
    CHAVE chaves[ORDEM-1];    // Chaves e respectivos ponteiros para registros
    struct pagina* filho[ORDEM]; // Ponteiros para os descendentes
} PAGINA;
```