

Árvores B - Inserção e Busca

Prof.: Leonardo Tórtoro Pereira
leonardop@usp.br

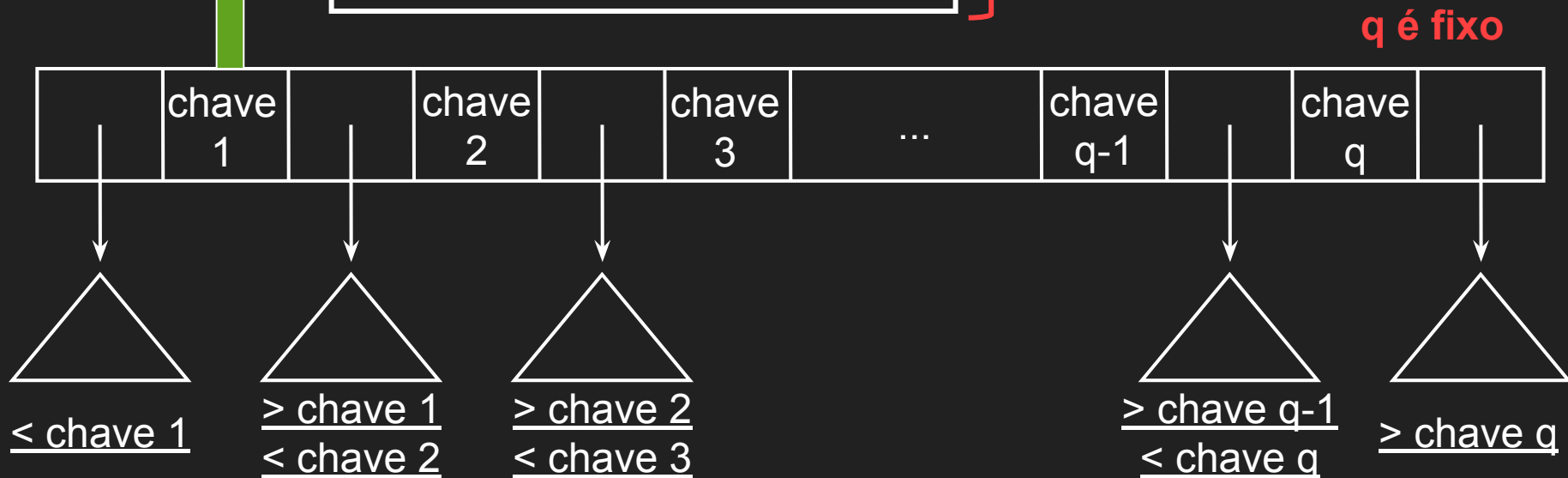
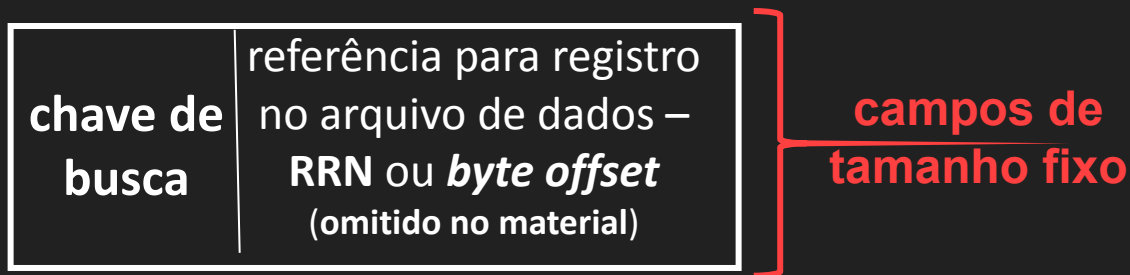
*Material baseado em aulas dos professores: Elaine Parros Machado de Souza, Gustavo Batista, Robson Cordeiro, Moacir Ponti Jr., Maria Cristina Oliveira e Cristina Ciferri.

O que sabemos de Árvores-B?

Relembrando...

- Definição
- Estrutura de Nós
- Lógica de Implementação
- Propriedades
- Cálculo de Profundidade
- Número de Acessos a Disco

Estrutura do Nó

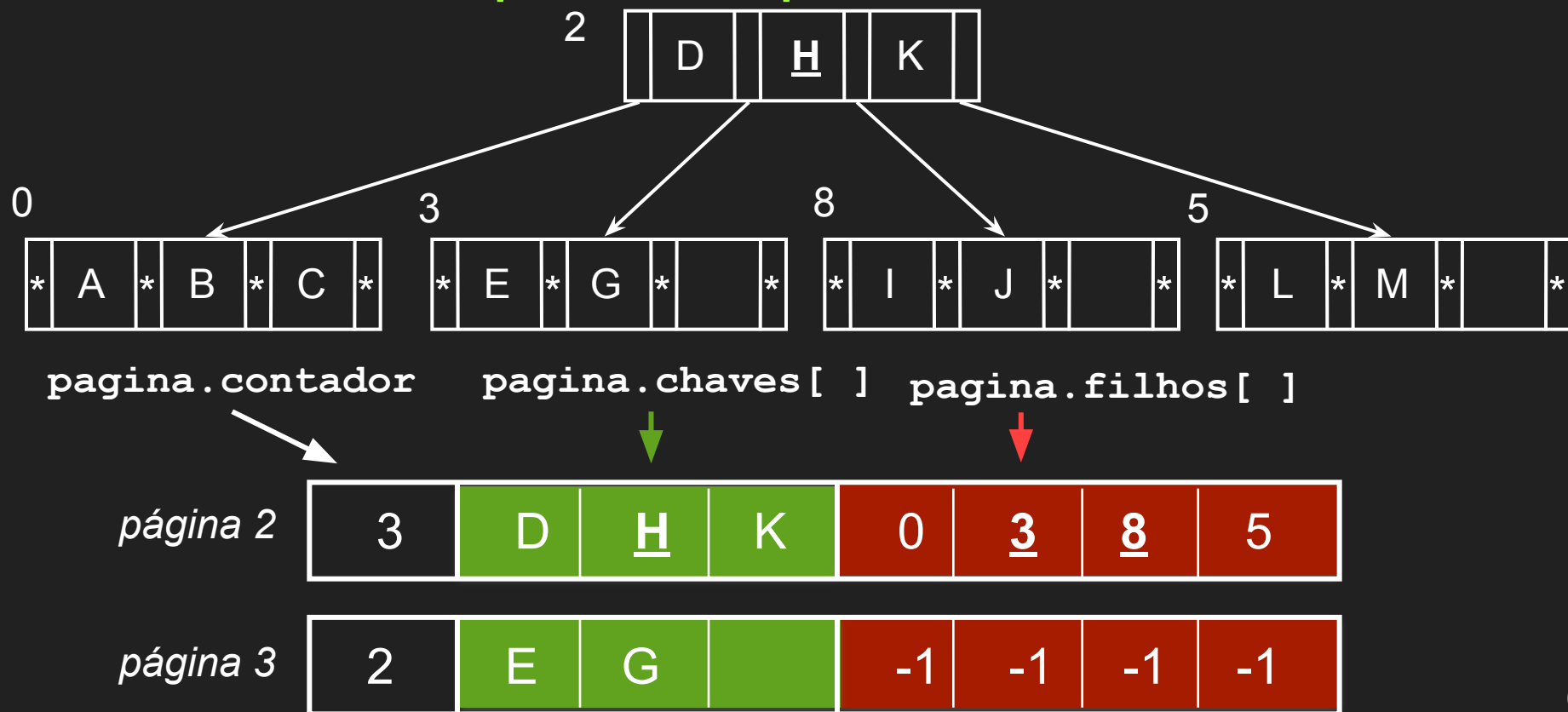


Exemplo de implementação de nó

```
#define ordem X /* X é calculado considerando o tamanho de  
página de disco do sistema, o tamanho da chave e dos itens  
de armazenados no nó */
```

```
typedef struct pagina {  
    int contador; //nro de chaves armazenadas  
    char chaves[ordem-1]; //assumindo chaves char  
    int filhos[ordem]; // armazena o RRN dos filhos  
    bool folha;  
} PAGINA;
```

Exemplo de arquivo na árvore B



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

Número máximo de acessos a disco para encontrar qualquer chave!

Inserção de Chaves

Inserção de Chaves

→ Inserção

- ◆ Sempre realizada nos nós folha

→ Situações a serem analisadas

- ◆ Árvore vazia (situação inicial)
- ◆ Overflow no nó raiz
- ◆ Inserção em nós folha
 - Sem overflow
 - Com overflow

Inserção - Árvore Vazia

Inserção - Árvore Vazia

- Criação e preenchimento do nó raiz
 - ◆ Primeira chave (árvore vazia)
 - Criação do nó raiz
 - ◆ Demais chaves
 - Inserção até a capacidade limite do nó
 - ◆ Chaves ordenadas

Inserção - Árvore Vazia

- Exemplo:
 - ◆ Nó com capacidade para 7 chaves
 - ◆ Ordem 8
- Chaves: letras do alfabeto
- Situação inicial: árvore vazia

Inserção - Árvore Vazia

- Chaves B A C E F D G
 - ◆ Inseridas aleatoriamente
 - ◆ Mantidas ordenadas no nó
 - Procedimento simplificado
 - A página é lida, a chave é inserida com inserção ordenada em RAM, página é re-escrita em disco
- Nó raiz = nó folha nesse momento
- Ponteiros (*)
 - ◆ -1 ou fim de lista (NULL)

Exemplo

Inserção - Árvore Vazia

Inserção de B:



página 1



Inserção - Árvore Vazia

Inserção de A:



página 1



Inserção - Árvore Vazia

Ao final das inserções:

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

página 1

7	A	B	C	...	G	-1	-1	...	-1
---	---	---	---	-----	---	----	----	-----	----

Inserção - Overflow nó Raiz

Inserção - Overflow nó Raiz

- Passo 1 – particionamento do nó (split)
 - ◆ Nó original
 - Nó original + novo nó
 - Split “1-to-2” ou “two-way”
 - ◆ Chaves (incluindo nova) são distribuídas uniformemente nos dois nós

Exemplo

Inserção - Overflow nó Raiz

Tínhamos esta árvore:



página 1



Inserção - Overflow nó Raiz

Inserção de J:



Inserção - Overflow nó Raiz

- Passo 1 – particionamento do nó (split)
 - ◆ Nó original
 - Nó original + novo nó
 - Split “1-to-2” ou “two-way”
 - ◆ Chaves (incluindo nova) são distribuídas uniformemente nos dois nós

Inserção - Overflow nó Raiz

Inserção de J:



página 1



página 2



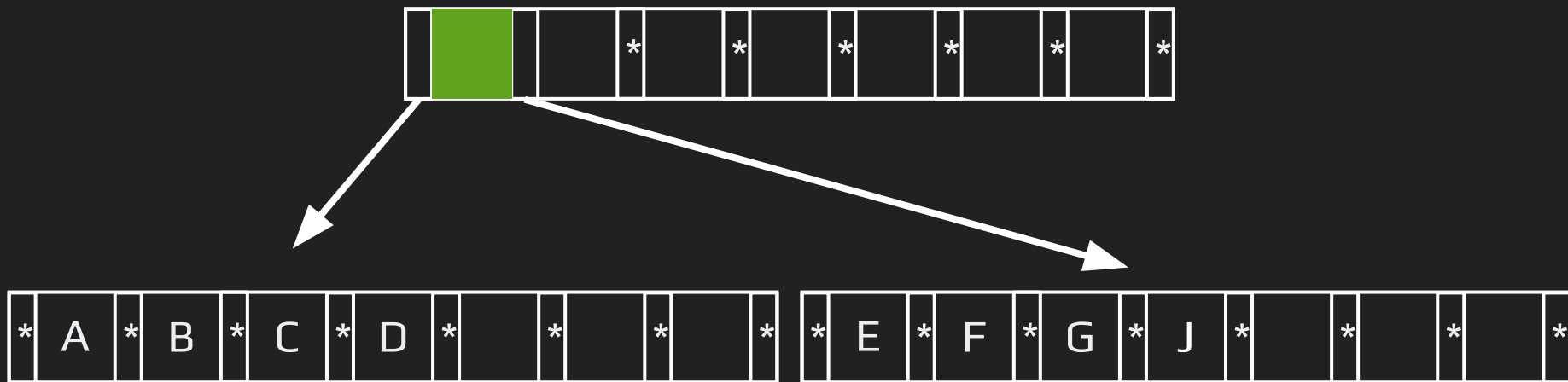
Inserção - Overflow nó Raiz

- Passo 2 – criação de uma nova raiz
 - ◆ A criação de um nível mais alto na árvore permite a escolha da chave separadora

Inserção - Overflow nó Raiz

Nova raiz será construída com 1 elemento

Qual deve ser incluído?



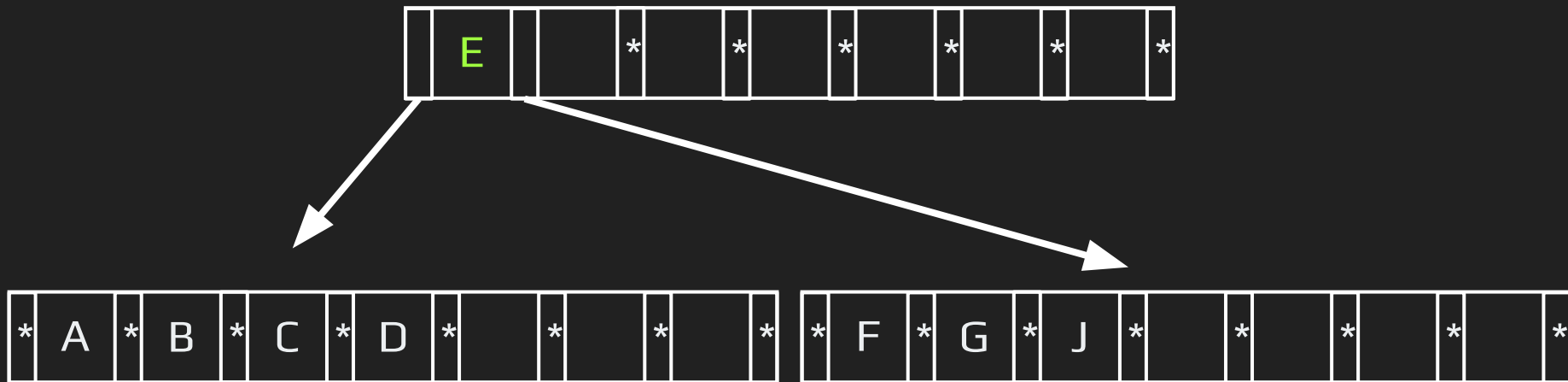
Inserção - Overflow nó Raiz

- Passo 3 – Promoção de chave (promotion)
 - ◆ A primeira chave do novo nó após particionamento é promovida para o nó raiz

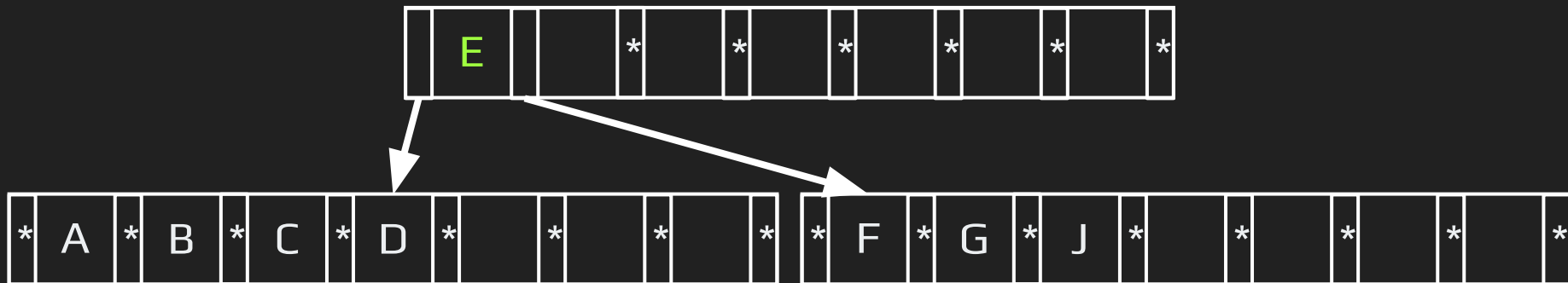
Inserção - Overflow nó Raiz

Nova raiz será construída com 1 elemento

Qual deve ser incluído?



Inserção - Overflow nó Raiz



página 1	4	A	B	C	D	...	-1	-1	...	-1
página 2	3	F	G	H		...	-1	-1	...	-1
página 3	1	E				...	1	2	...	-1

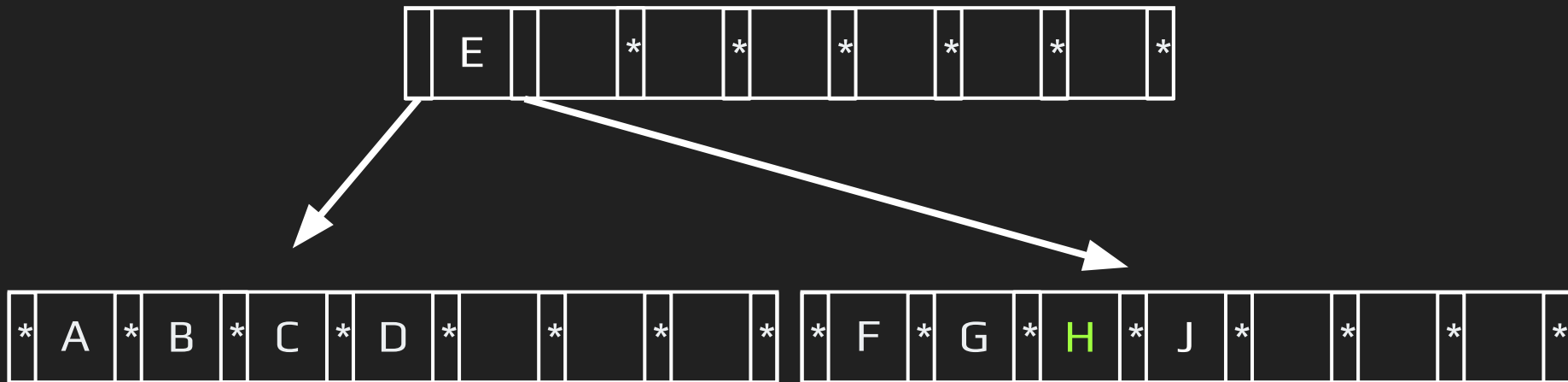
Inserção - Nós Folha

Inserção - Nós Folha

- Passo 1 – Pesquisa (Vamos ver em seguida como é)
 - ◆ Árvore é percorrida até encontrar o nó folha no qual a nova chave será inserida
 - Páginas são lidas para memória principal

Inserção - Nós Folha

- Passo 2 – inserção em nó com espaço (sem overflow)
- ◆ Ordenação das chaves após a inserção
- ◆ Ex: inserção da chave H



Exemplo

Inserção - Nós Folha

- Passo 2 – inserção em nó cheio (com overflow)
 - ◆ Particionamento (split)
 - Criação de um novo nó
 - Nó original \Rightarrow nó original + novo nó
 - Distribuição uniforme das chaves nos dois nós

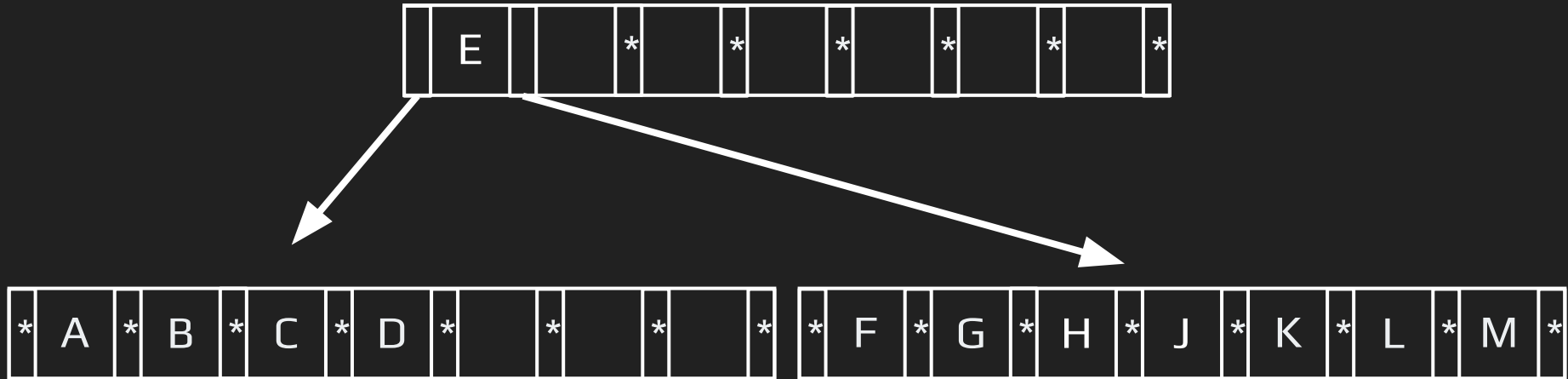
Inserção - Nós Folha

→ Passo 2 – inserção em nó cheio (com overflow)

◆ Promoção

- Escolha da primeira chave do novo nó como chave separadora no nó pai
 - Nó por onde a pesquisa passou antes
- Reordenação e ajuste do nó pai para apontar para o novo nó
- Propagação de overflow

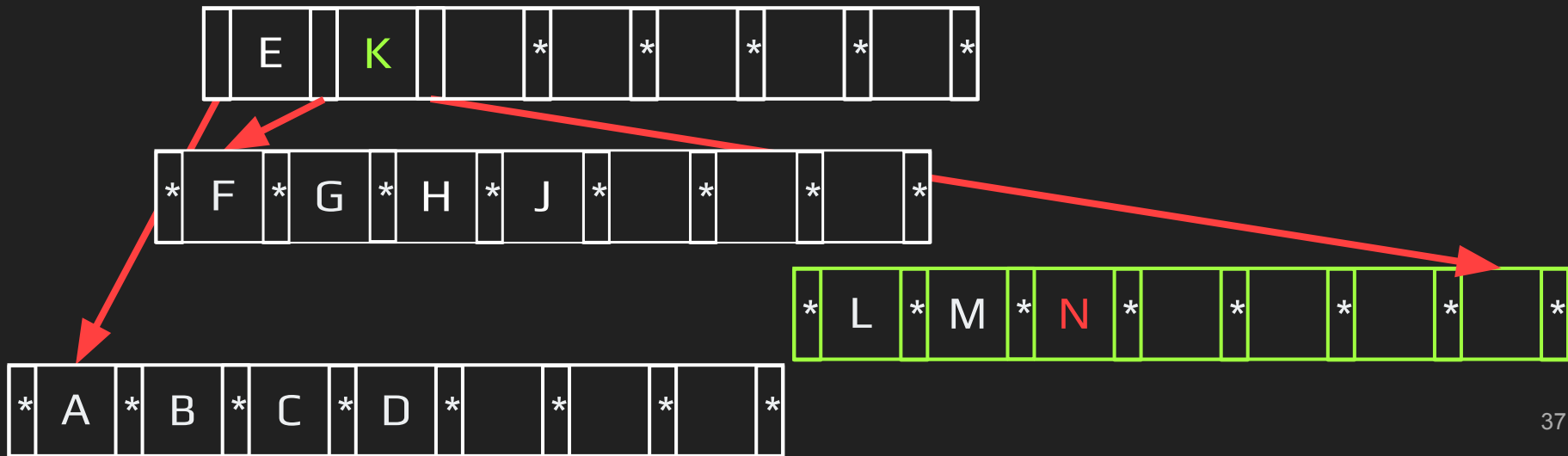
Inserção - Nós Folha



Inserção - Nós Folha

→ Inserção de N

- ◆ Particionamento: Criação de Nova Página
- ◆ Promoção: Inserção de nova chave no nó pai



Exemplo

Exemplo

Exemplo

- Insira as seguintes chaves em um índice árvore-B
 - ◆ C S D T A M P I B W N G U K E H O L J Y Q Z F X V
- Ordem da árvore-B: 4
 - ◆ Em cada nó (página de disco)
 - Número de chaves: 3
 - Número de ponteiros: 4

Exemplo

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

→ Inserção de C, S, D

◆ Criação do nó raiz

0

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

página 0

3	C	D	S	-1	-1	-1	-1
---	---	---	---	----	----	----	----

Exemplo

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

- Inserção de T
- ◆ Nó raiz cheio
- Split do nó 0
- ◆ Cria nó 1



página 0



página 1



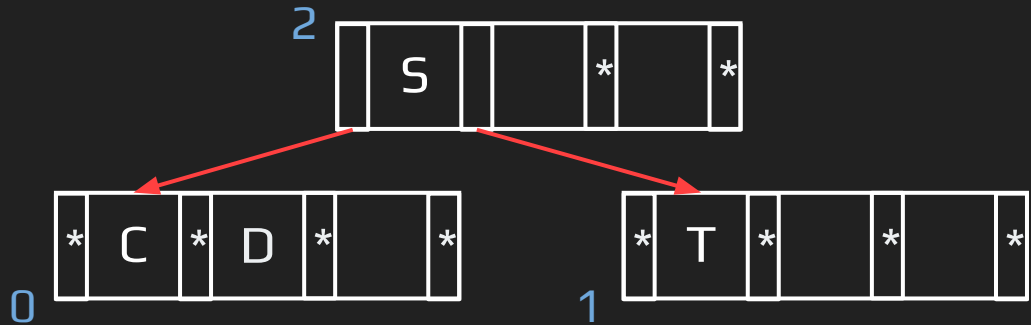
Exemplo

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

→ Cria uma nova raiz

◆ Nó 2

→ Promove S



página 0

2	C	D		-1	-1	-1	-1
---	---	---	--	----	----	----	----

página 1

1	T			-1	-1	-1	-1
---	---	--	--	----	----	----	----

página 2

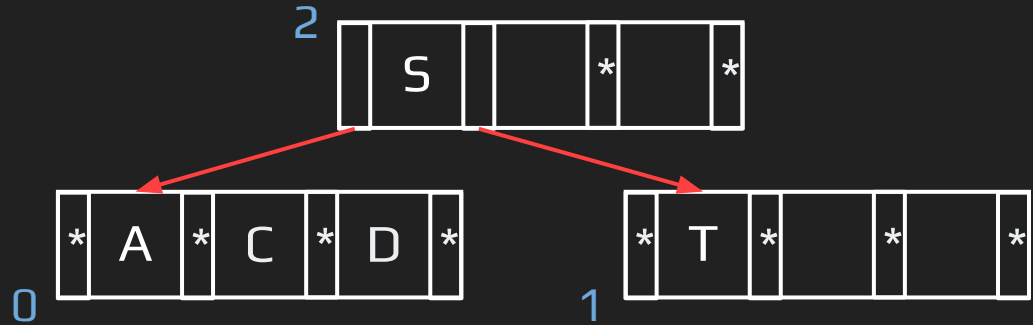
1	S			0	1	-1	-1
---	---	--	--	---	---	----	----

Exemplo

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

→ Inserção de A

◆ Nó folha 0 com espaço



página 0

3	A	C	D	-1	-1	-1	-1
---	---	---	---	----	----	----	----

página 1

1	T			-1	-1	-1	-1
---	---	--	--	----	----	----	----

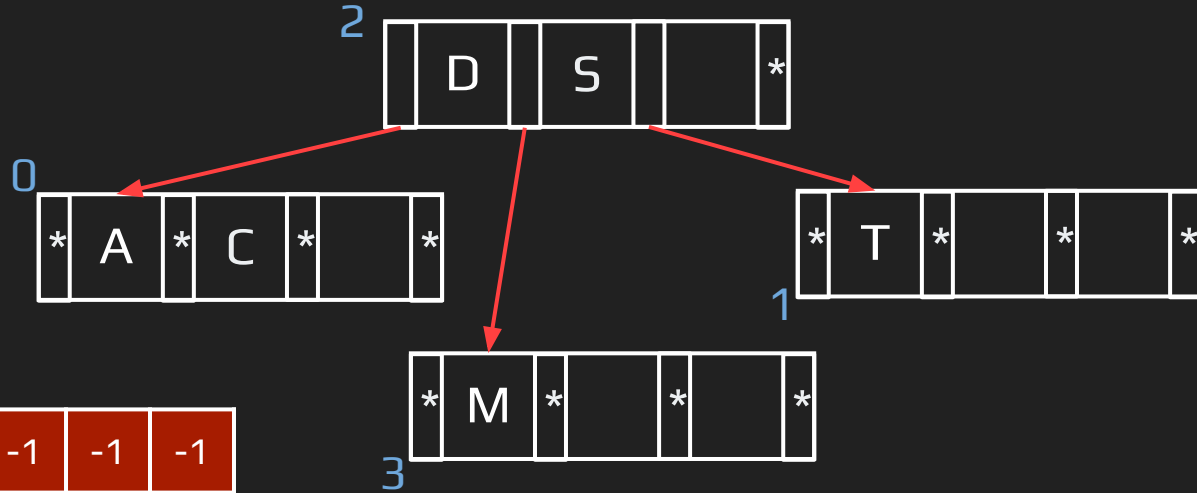
página 2

1	S			0	1	-1	-1
---	---	--	--	---	---	----	----

Exemplo

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

- Inserção de M
 - ◆ Nó **folha 0** cheio
- Split do nó 0
 - ◆ Criação do nó 3
- Promoção de D



página 0

2	A	C		-1	-1	-1	-1
---	---	---	--	----	----	----	----

página 1

1	T			-1	-1	-1	-1
---	---	--	--	----	----	----	----

página 2

2	D	S		0	3	1	-1
---	---	---	--	---	---	---	----

página 3

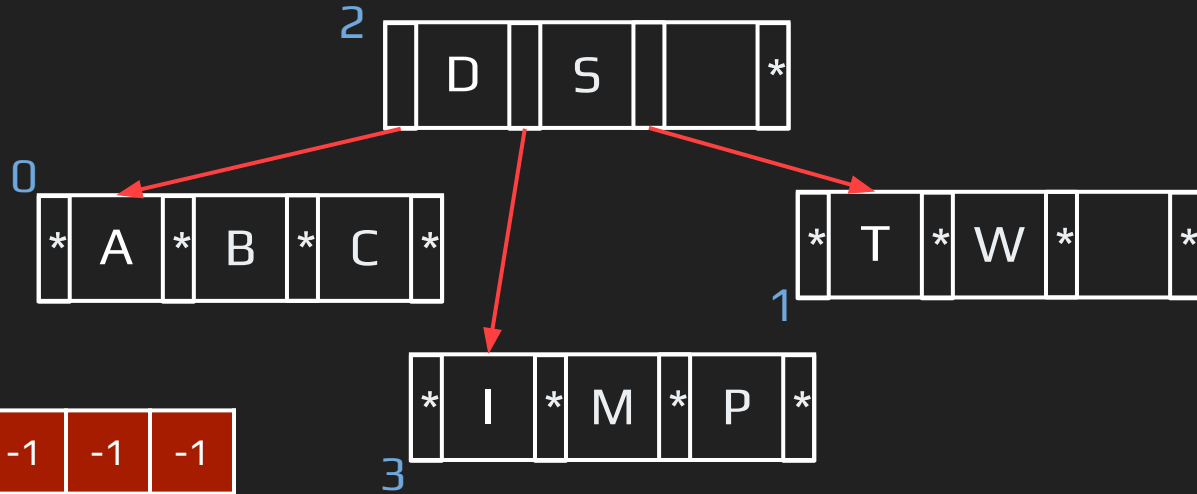
1	M			-1	-1	-1	-1
---	---	--	--	----	----	----	----

Exemplo

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

→ Inserção de P, I, B,
W

→ Nós folha com
espaço



página 0

3	A	B	C	-1	-1	-1	-1
---	---	---	---	----	----	----	----

página 1

2	T	W		-1	-1	-1	-1
---	---	---	--	----	----	----	----

página 2

2	D	S		0	3	1	-1
---	---	---	--	---	---	---	----

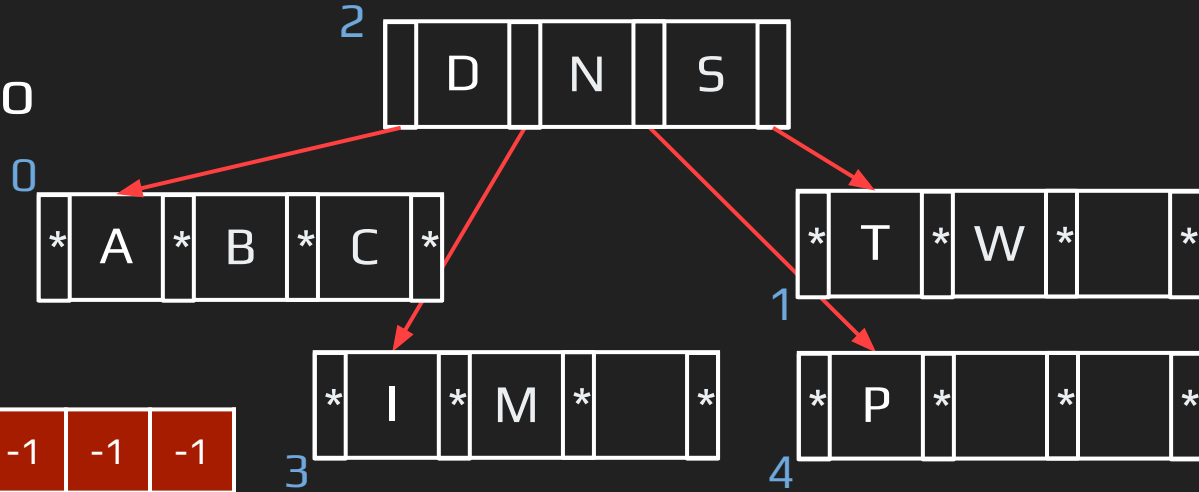
página 3

3	I	M	P	-1	-1	-1	-1
---	---	---	---	----	----	----	----

Exemplo

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

- Inserção de N
 - ◆ Nó folha 3 cheio
- Split do nó 3
 - ◆ Cria nó 4
- Promove N



página 0

3	A	B	C	-1	-1	-1	-1
---	---	---	---	----	----	----	----

página 1

2	T	W		-1	-1	-1	-1
---	---	---	--	----	----	----	----

página 2

3	D	N	S	0	3	4	1
---	---	---	---	---	---	---	---

página 3

2	I	M		-1	-1	-1	-1
---	---	---	--	----	----	----	----

página 4

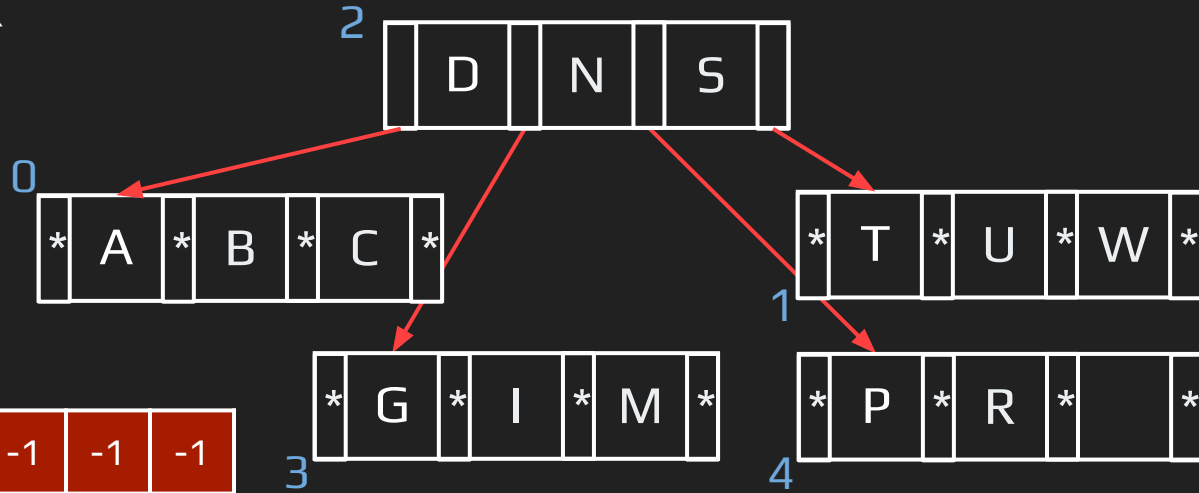
1	P			-1	-1	-1	-1
---	---	--	--	----	----	----	----

Exemplo

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

→ Inserção de G, U, R

◆ Nós folha com espaço



página 0

3	A	B	C	-1	-1	-1	-1
---	---	---	---	----	----	----	----

página 1

3	T	U	W	-1	-1	-1	-1
---	---	---	---	----	----	----	----

página 2

3	D	N	S	0	3	4	1
---	---	---	---	---	---	---	---

página 3

3	G	I	M	-1	-1	-1	-1
---	---	---	---	----	----	----	----

página 4

2	P	R		-1	-1	-1	-1
---	---	---	--	----	----	----	----

Exemplo

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

→ Inserção de K

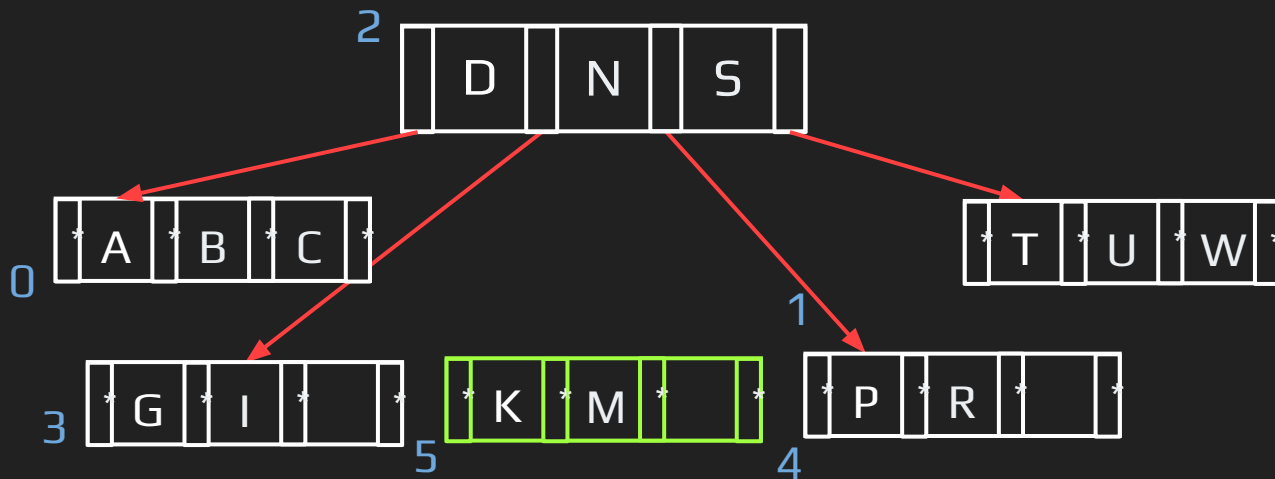
◆ Nó folha 3 cheio

◆ Split do nó 3

• Cria nó 5

◆ Propaga K

• Overflow



página 0

3	A	B	C	-1	-1	-1	-1
---	---	---	---	----	----	----	----

página 3

2	G	I		-1	-1	-1	-1
---	---	---	--	----	----	----	----

página 1

3	T	U	W	-1	-1	-1	-1
---	---	---	---	----	----	----	----

página 4

2	P	R		-1	-1	-1	-1
---	---	---	--	----	----	----	----

página 2

3	D	N	S	0	3	4	1
---	---	---	---	---	---	---	---

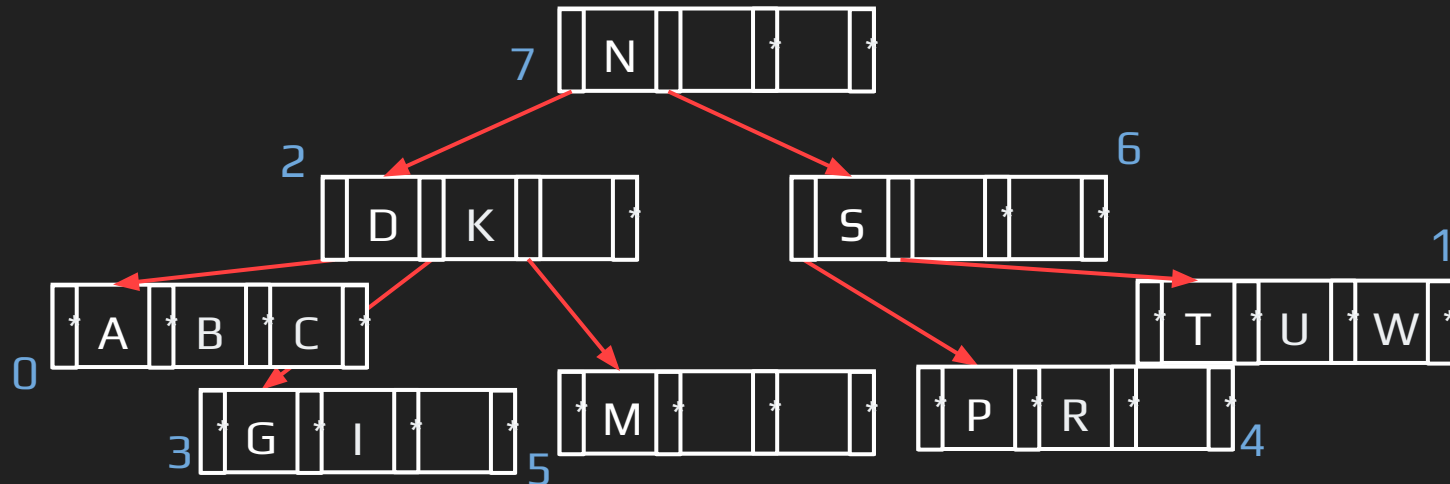
página 5

2	K	M		-1	-1	-1	-1
---	---	---	--	----	----	----	----

Exemplo

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

- Inserção de K
- ◆ Split do nó 2
 - Cria 6
 - ◆ Cria nova raiz
 - Nó 7
 - ◆ Promove N



página 0

3	A	B	C	-1	-1	-1	-1
---	---	---	---	----	----	----	----

página 1

3	T	U	W	-1	-1	-1	-1
---	---	---	---	----	----	----	----

página 2

2	D	K		0	3	5	-1
---	---	---	--	---	---	---	----

página 3

2	G	I		-1	-1	-1	-1
---	---	---	--	----	----	----	----

página 4

2	P	R		-1	-1	-1	-1
---	---	---	--	----	----	----	----

página 5

1	M			-1	-1	-1	-1
---	---	--	--	----	----	----	----

página 6

1	S			4	1	-1	-1
---	---	--	--	---	---	----	----

página 7

1	N			2	6	-1	-1
---	---	--	--	---	---	----	----

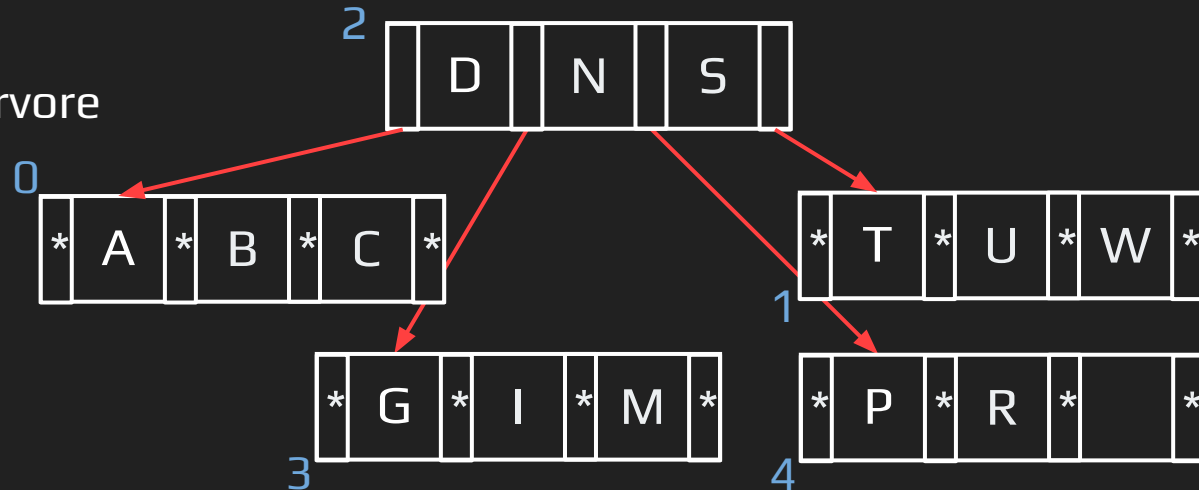
Finalizem a Construção da Árvore :)

Exemplo

Busca

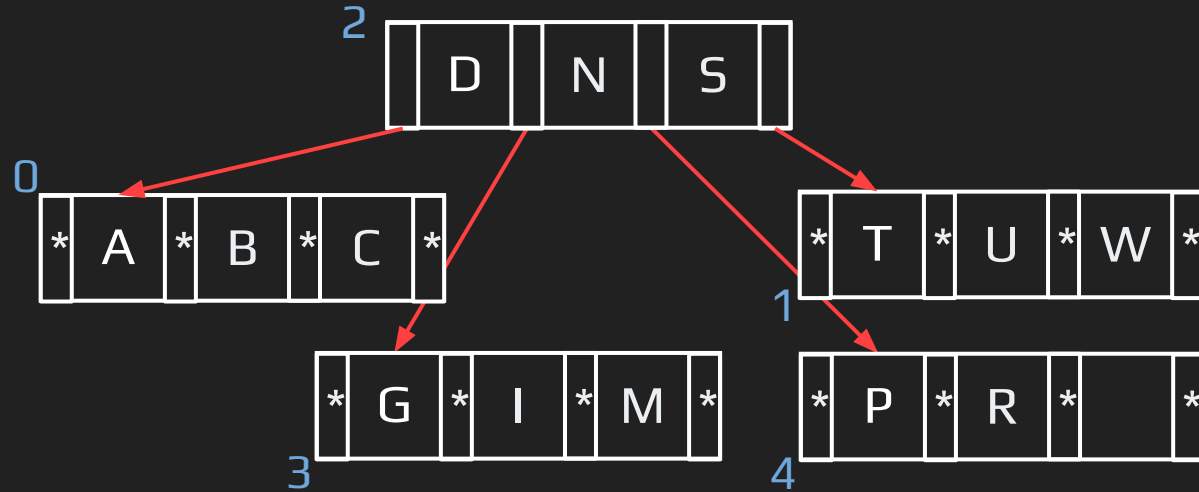
Busca

- Busca pela chave K
 - ◆ Trazer a página raiz (2) para a memória
 - ◆ Buscar chave na raiz
 - ◆ Se não encontrada
 - Buscar na subárvore apropriada



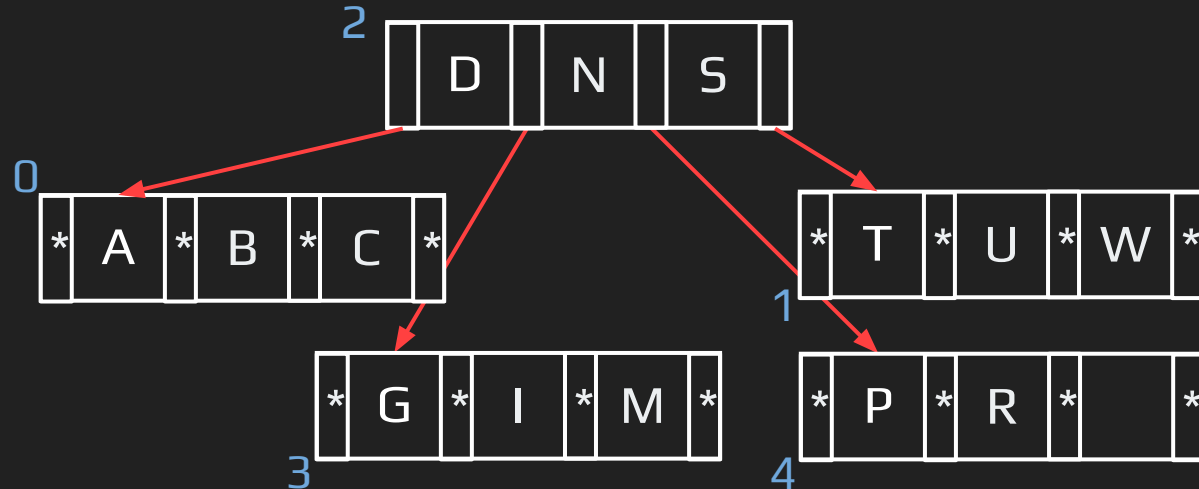
Busca

- Busca pela chave K
 - ◆ Trazer a página (3) para a memória
 - ◆ Buscar chave no nó
 - ◆ Se não encontrada
 - Fim



Busca

- Complexidade no pior caso?
 - ◆ Quantos acessos a Disco?
 - Profundidade da árvore :)



Busca

- $d \leq 1 + \log_{\lceil m/2 \rceil}(N/2)$
 - ◆ $O(\log_{\lceil m/2 \rceil}(N))$
- Exemplo
 - ◆ 1M chaves
 - ◆ Árvore de ordem 512
 - ◆ $d \leq 1 + \log_{256}(500.000)$
 - $d \leq 3.37$
- No máximo, 3 acessos a disco para qualquer chave

Busca

- Algoritmos recursivos
 - ◆ Dois estágios de processamento
 - Em páginas inteiras
 - Dentro das páginas
- Estrutura de dados
 - ◆ Determina cada página de disco
 - ◆ Pode ser implementada de diferentes maneiras

Busca

- Implementação vista aula passada...
- ◆ Contador de ocupação
- ◆ Chave
- ◆ “ponteiros” para as páginas filhas

Busca

→ Uma possível implementação da página...

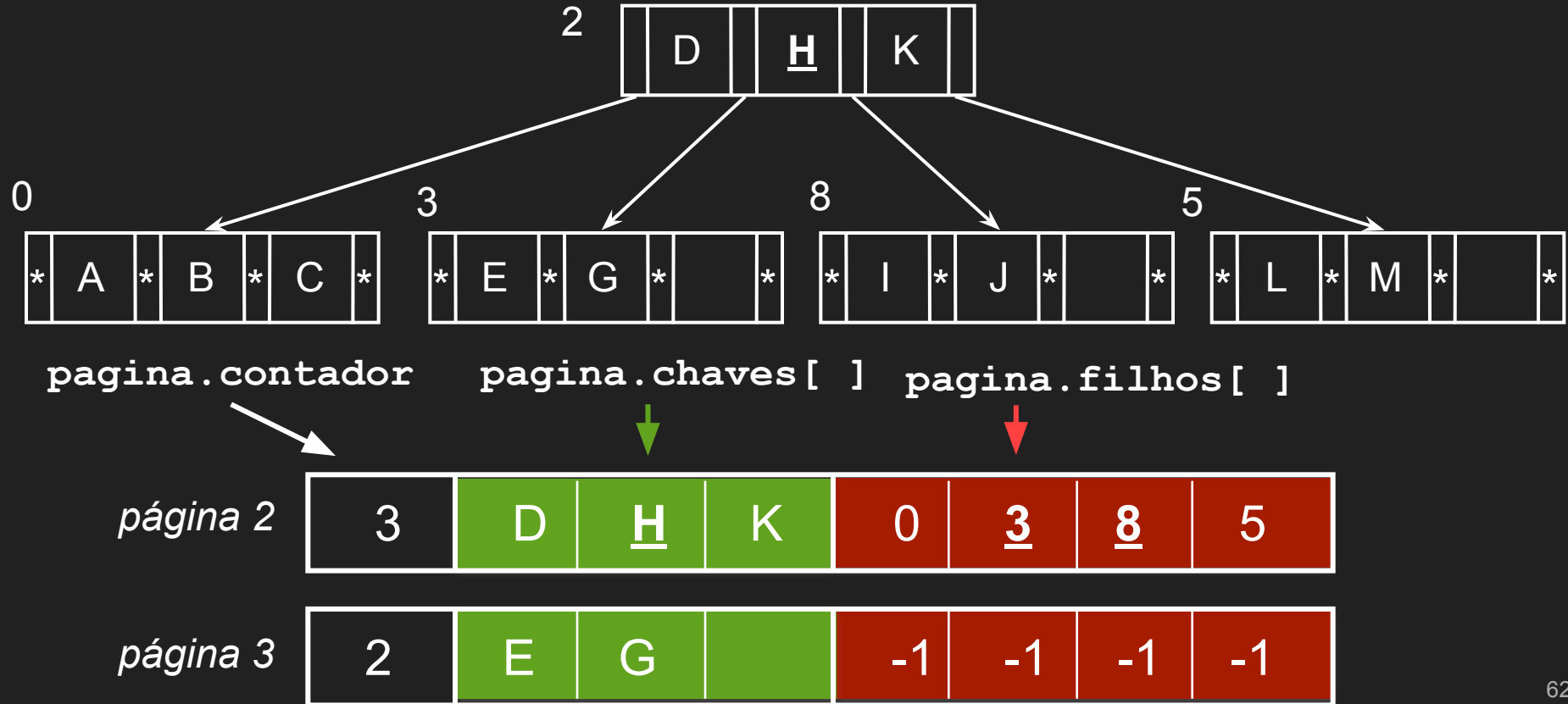
```
#define ordem 8
```

```
typedef struct pagina {  
    int contador;  
    char chaves[ordem-1]; //assumindo chaves char  
    int filhos[ordem]; //armazena o RRN dos filhos  
    bool folha;  
} PAGINA;
```

Busca

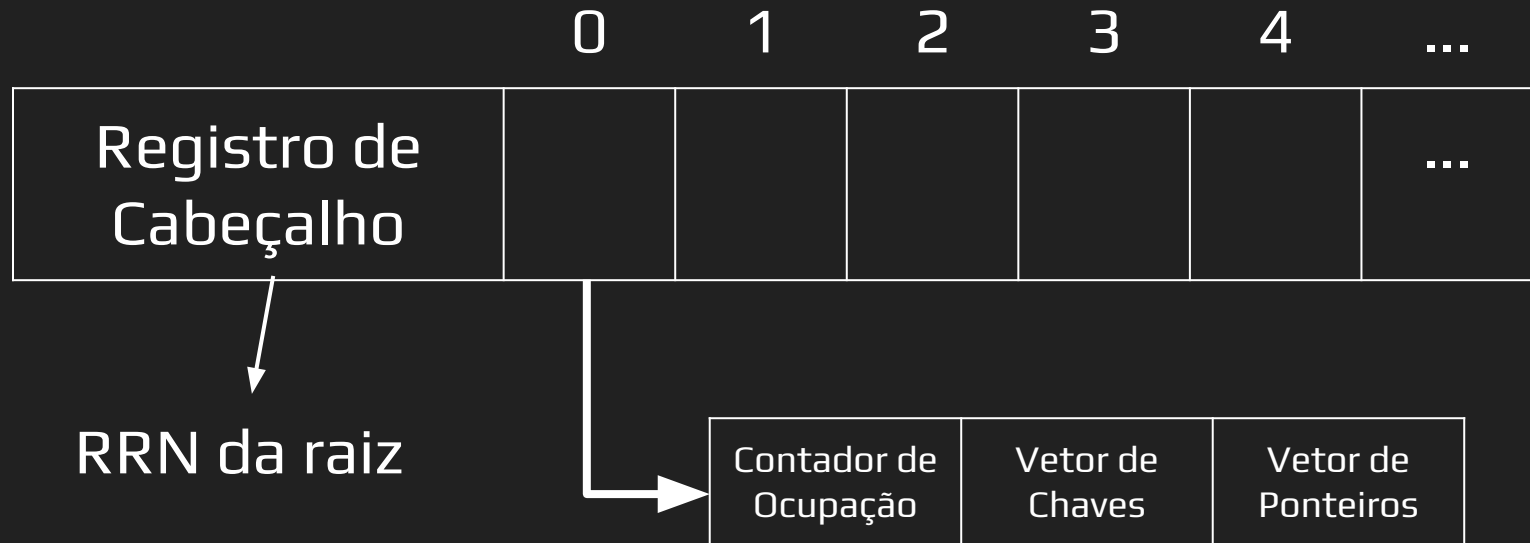
- Para indexação
 - ◆ Cada chave tem uma referência para o registro de dados armazenada

Exemplo de arquivo na árvore B



Busca

- Conjunto de registros de tamanho fixo
- ◆ Cada registro corresponde a uma página de disco



Busca

```
função busca(RRN,          //página atual sendo pesquisada
             chave,        //chave sendo procurada
             RRN_encontrado, //retorna a página que contém a chave
             pos_encontrada) //retorna posição da chave na página

se (RRN == -1) então
    retorne FALSO          //chave de busca não encontrada

senão
    leia página P identificada por RRN
    procure chave em P, e atribua a POS a posição do vetor onde a
                                chave deveria ocorrer

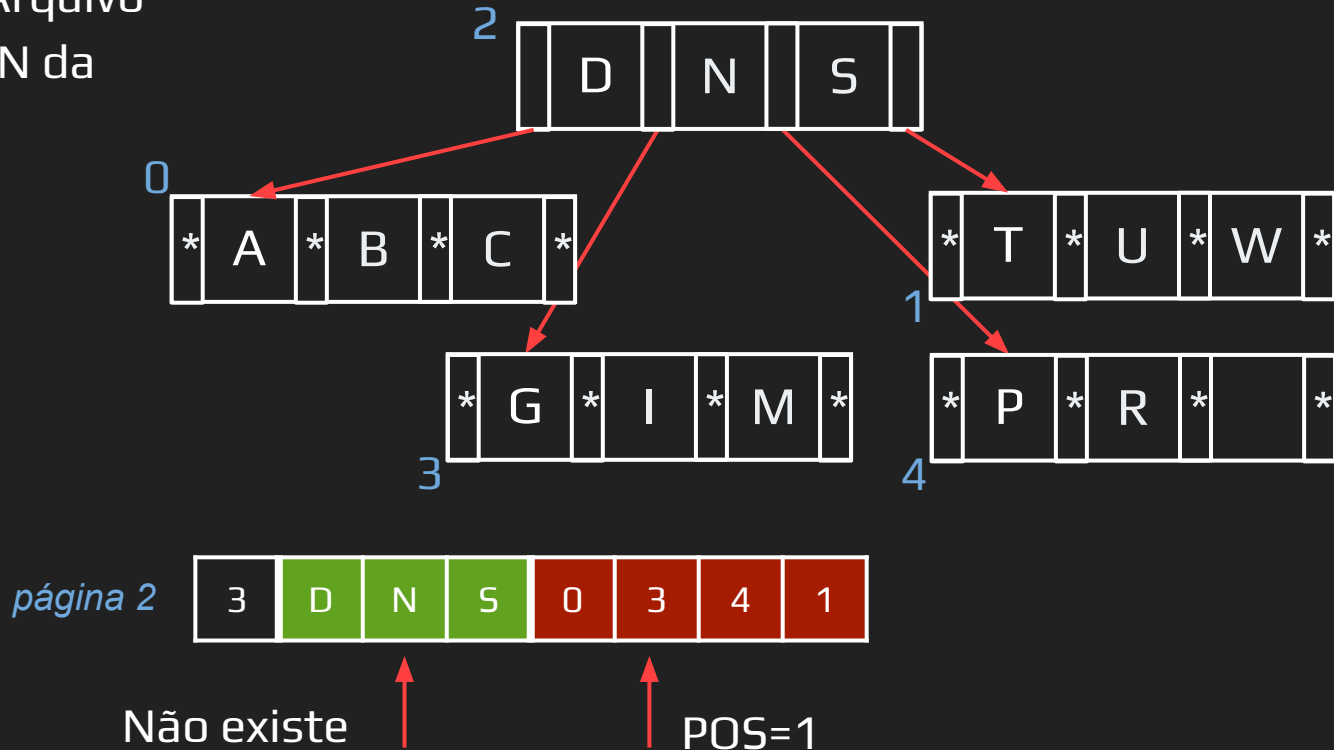
    se (chave_encontrada) então
        RRN_encontrado = RRN      //RRN atual - contém a chave
        pos_encontrada = POS      //POS - posição da chave na página
        retorne VERDADEIRO

    //se chave não encontrada, recomeça-se busca no filho apropriado
    senão
        retorne busca(P.filhos[POS], chave, RRN_encontrado,
                       pos_encontrada)
```


Busca Chave K

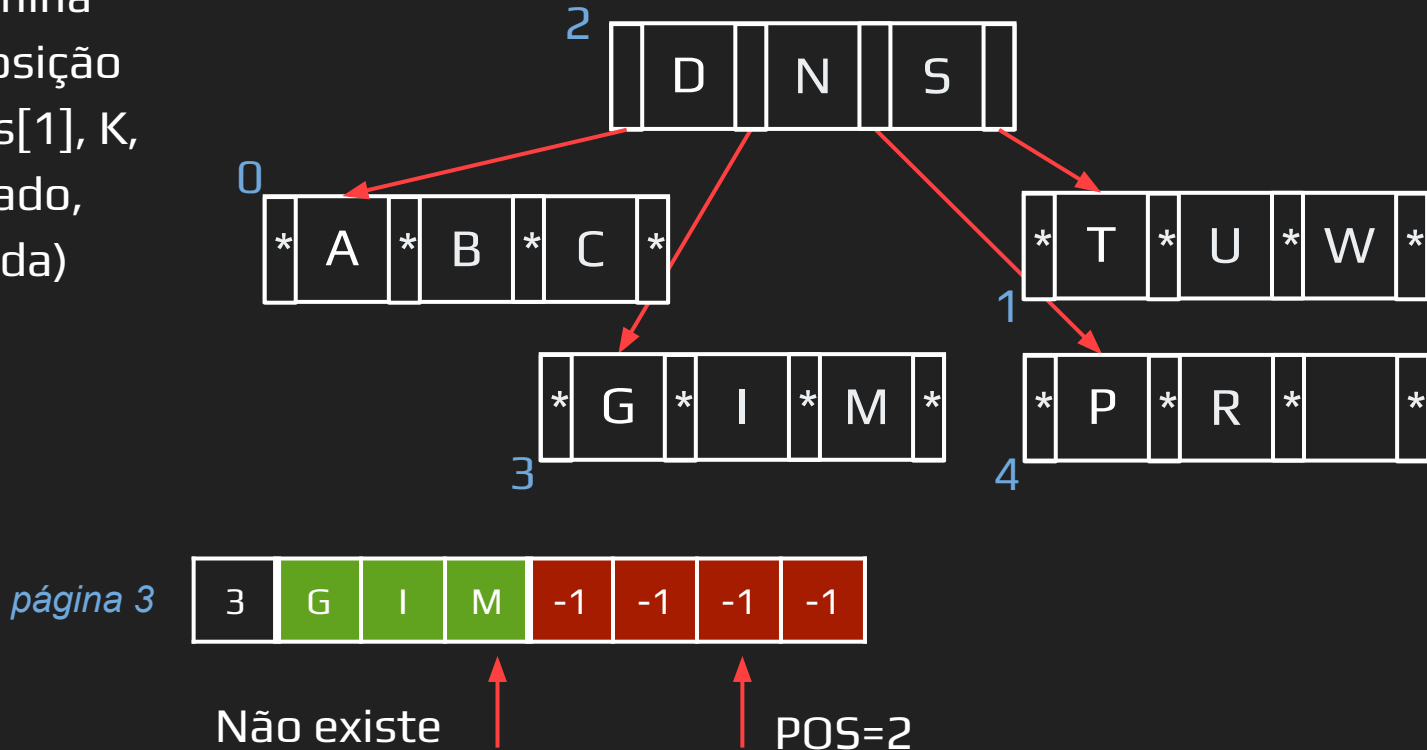
Busca da Chave K

- Ler cabeçalho do Arquivo para encontrar RRN da raiz
- Busca (2, K, RRN_encontrado, pos_encontrada)



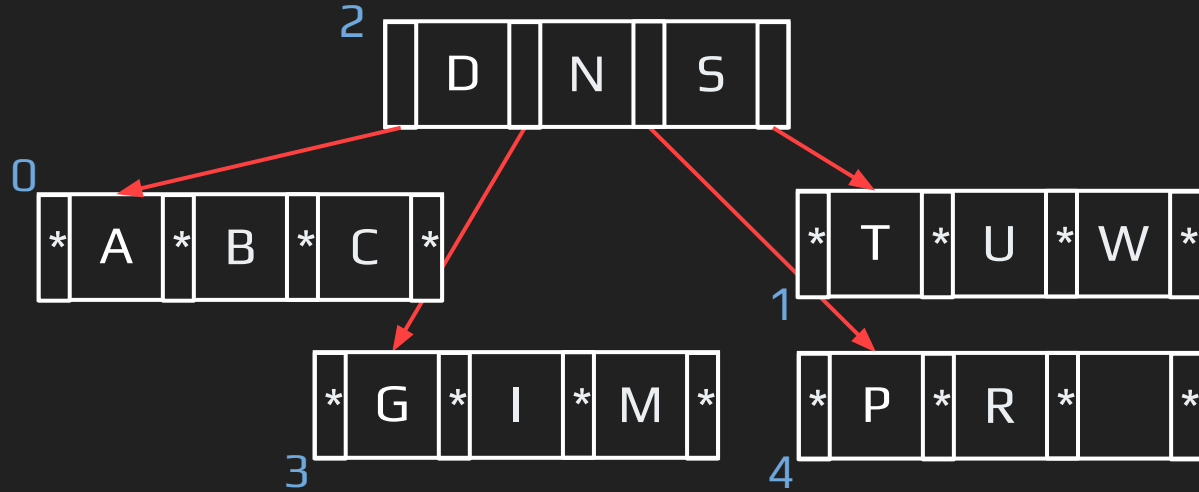
Busca da Chave K

- Busca página filha indicada na posição
- Busca (P.filhos[1], K, RRN_encontrado, pos_encontrada)



Busca da Chave K

- Busca (P.filhos[2], K, RRN_encontrado, pos_encontrada)
- P.filhos[2] = -1
- Chave de busca não encontrada
- Retorna FALSO

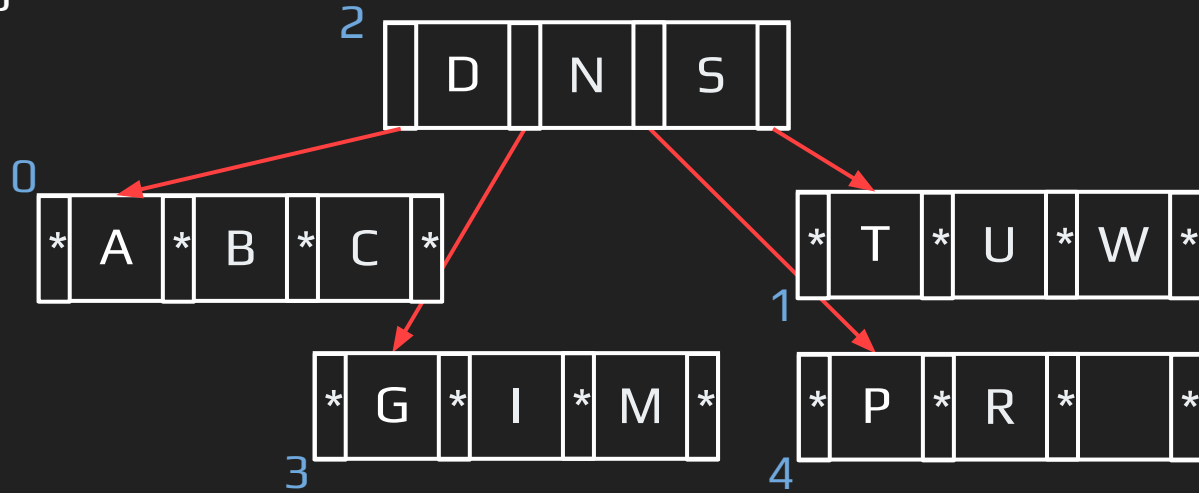


Exemplo

Busca Chave M

Busca da Chave M

- Ler cabeçalho do Arquivo para encontrar RRN da raiz
- Busca (2, M, RRN_encontrado, pos_encontrada)



página 2

3	D	N	S	0	3	4	1
---	---	---	---	---	---	---	---

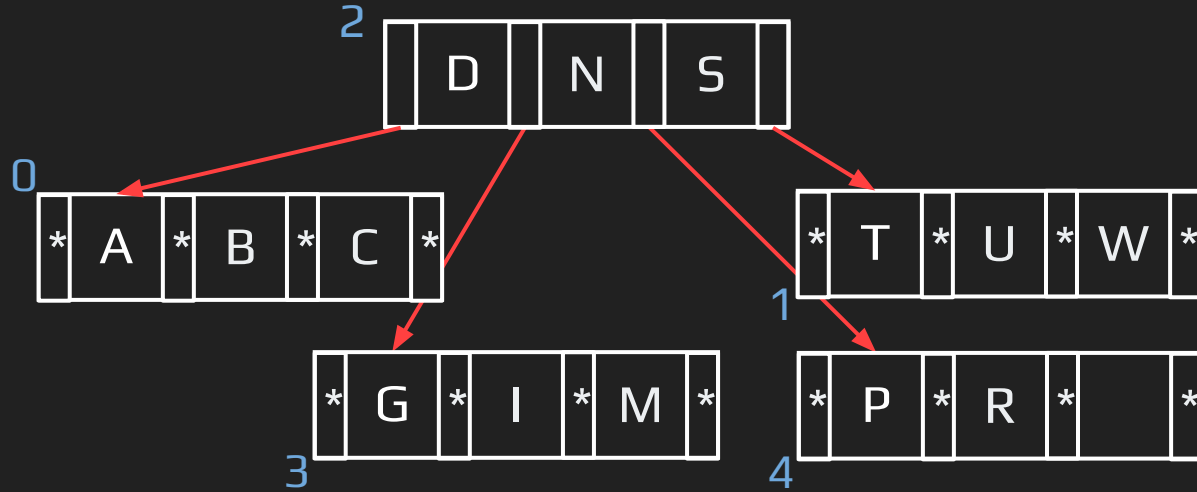
Não existe

POS=1

Busca da Chave K

- Busca página filha indicada na posição
- Busca (P.filhos[1], M, RRN_encontrado, pos_encontrada)
- Chave encontrada

pos_encontrada = 2
RRN_encontrado = 3
Retorna Verdadeiro



página 3

3	G	I	M	-1	-1	-1	-1
---	---	---	---	----	----	----	----

Está aqui!

Exemplo

Referências

- M. J. Folk and B. Zoellick, File Structures: A Conceptual Toolkit, Addison Wesley, 1987.