

Algoritmos e Estrutura de Dados II (AE23CP-3CP)

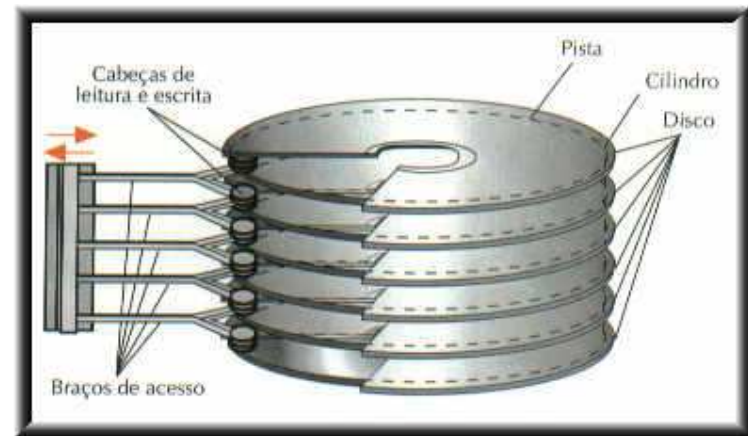
Aula #13 - Árvores B

Prof^a Luciene de Oliveira Marin
lucienemarin@utfpr.edu.br



Pesquisa em Memória Primária

- Pesquisa Sequencial
- Pesquisa Binária
- Árvores de Pesquisa
 - Árvore Binária de Pesquisa (com e sem balanceamento)
 - AVL
 - Árvore Vermelha-preta (*Red-black Tree*)
- Tabela Hashing

Pesquisa em Memória Secundária



Pesquisa em Memória Secundária

Memória Principal	Memória Secundária
Acesso mais rápido Mais Cara	Acesso mais lento Mais barata
	

Árvores B (B-Tree)

Contexto

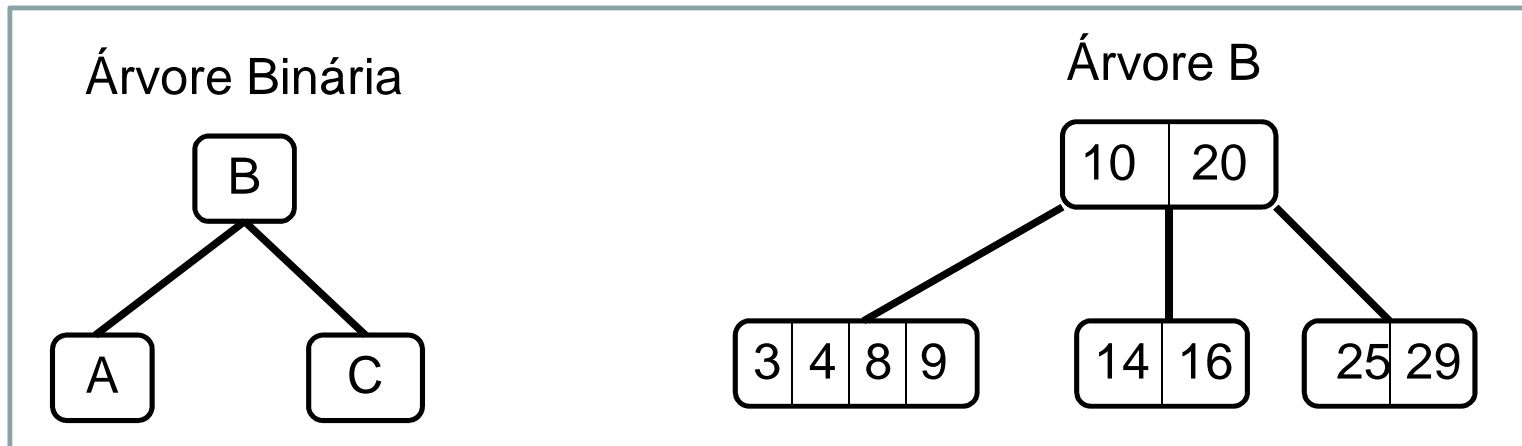
- São largamente utilizadas como forma de armazenamento em memória secundária;
- Diversos sistemas comerciais de banco de dados as empregam;
- Sistemas de arquivos em sistemas operacionais também as empregam.

Definição

- Um nó em uma árvore B é também chamado de **página**;
- Implica dizer que: um nó (página) em uma árvore B pode ter mais de um elemento.
- Ordem de uma árvore B:
 - Cormen *et. al.*: definem ordem como o número mínimo de filhos que uma árvore pode ter.
 - Knuth propôs que a ordem fosse o número máximo de páginas (nós) filhas que toda página pode conter.
- Obs.: para estes slides vamos considerar a definição de Knuth.

Definição

- Exemplo



Árvore Binária é
uma árvore de
ordem $N = 2$

Árvore B é uma árvore N-ária
Neste exemplo $N = 5$

Observação

Função Teto e Função Piso:

- A função $\lceil \cdot \rceil$ é chamada de função **teto**: se x é um número real qualquer, então $\lceil x \rceil$ corresponde ao menor inteiro maior ou igual a x .

$$\text{Ex.: } \lceil 1,6 \rceil = 2, \quad \lceil 2,6 \rceil = 3, \quad \lceil -2.71 \rceil = -2$$

- A função $\lfloor \cdot \rfloor$ é chamada de função **piso**: se x é um número real qualquer, então $\lfloor x \rfloor$ corresponde ao maior inteiro menor ou igual a x .

$$\text{Ex.: } \lfloor 1,6 \rfloor = 1, \quad \lfloor 2,6 \rfloor = 2, \quad \lfloor -2,71 \rfloor = -3$$

Definição Árvore B

- **Raiz:**
 - Elementos: mínimo 1 e máximo $N-1$
 - Subárvores: mínimo duas e no máximo N ;
- **Nó interno:** (que não é raiz nem folha):
 - Elementos: mínimo $\lceil N/2 \rceil - 1$ e Máximo $N-1$
 - Subárvores: mínimo $\lceil N/2 \rceil$ e Máximo N
- **Folhas:**
 - Todas as folhas estão no mesmo nível.

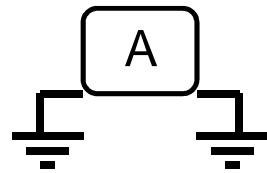
Atenção!

- Atentar para a diferença entre nós:
- Árvore Binária: cada nó tem um elemento e entre zero (no caso de ser um nó folha) e no máximo dois filhos.
- Árvore n -ária: cada nó pode ter no máximo $n-1$ elementos e no máximo n filhos.

Exemplo

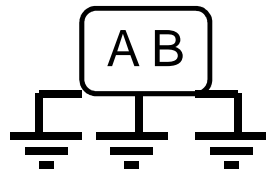
- Raiz:
 - Elementos: mínimo 1 e máximo $N-1$
 - Subárvores: mínimo duas e no máximo N ;
- Exemplos: Árvore de Ordem $N=5$

1 elemento



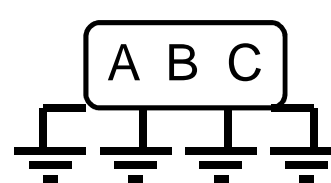
2 subárvores

2 elementos



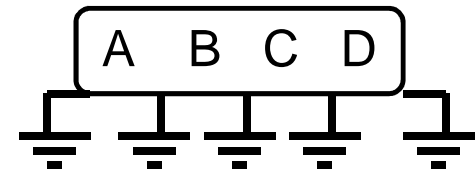
3 subárvores

3 elementos



4 subárvores

4 elementos

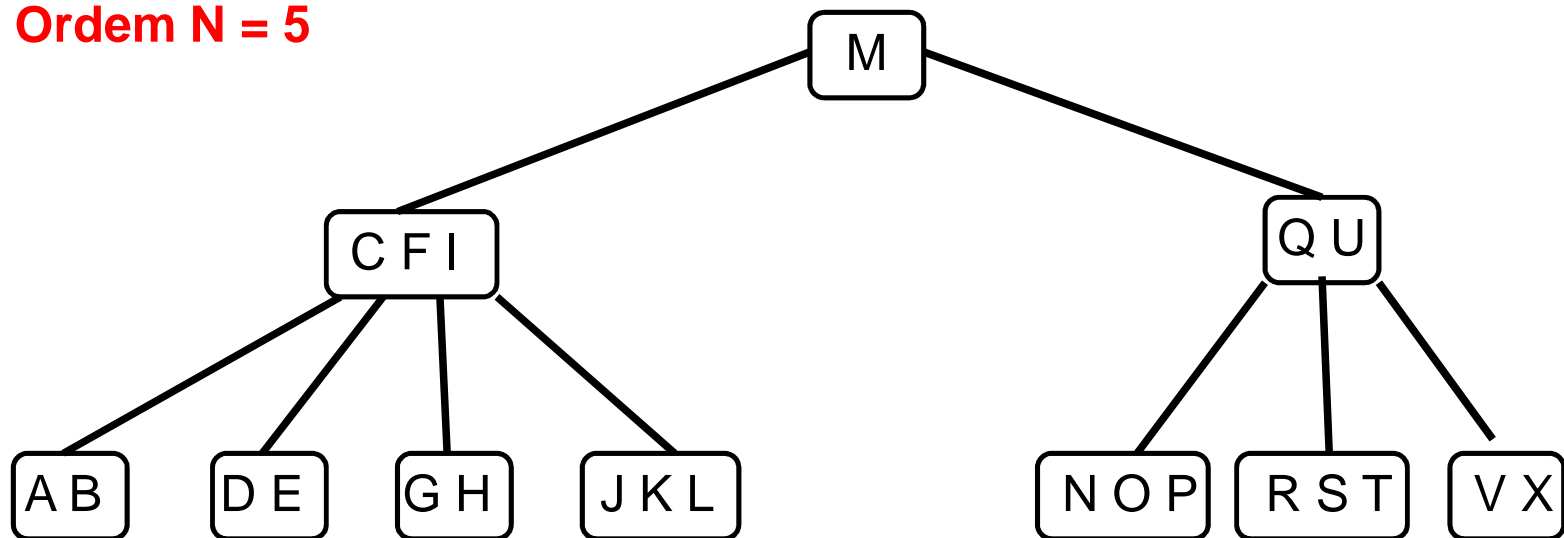


5 subárvores

Exemplo

- Nó interno (que não é raiz nem folha):
 - Elementos: mínimo $\lceil N/2 \rceil - 1$ e Máximo $N-1$
 - Subárvores: mínimo $\lceil N/2 \rceil$ e Máximo N

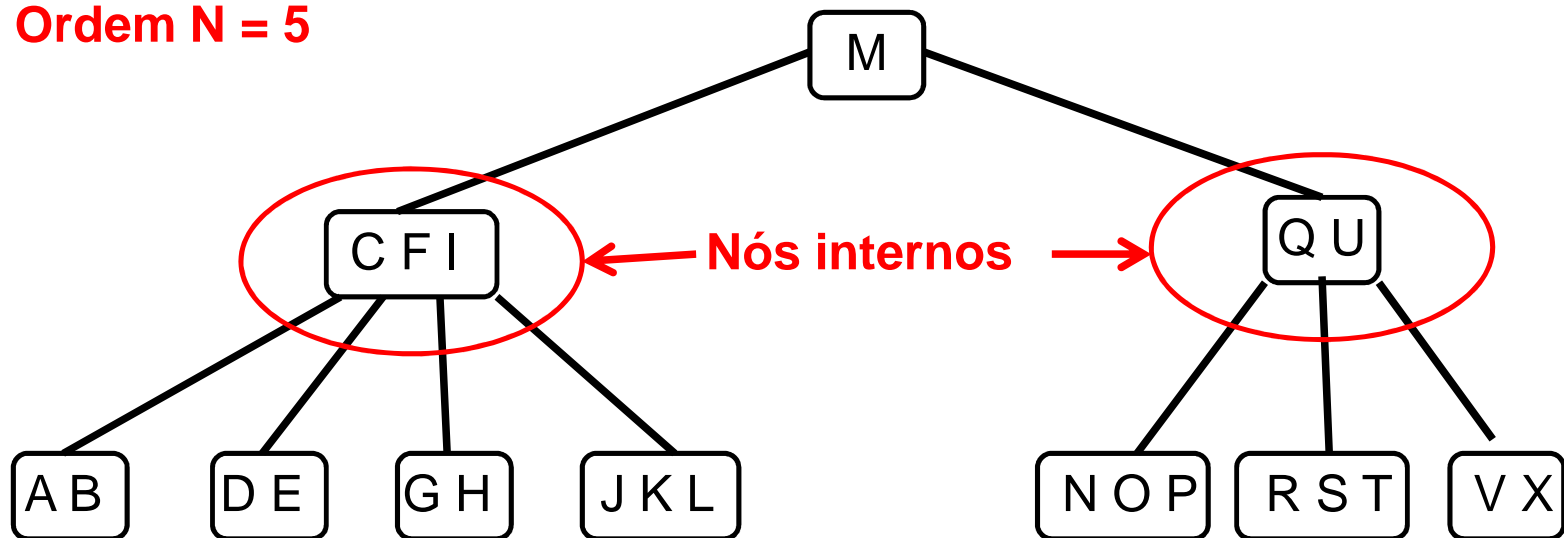
Ordem $N = 5$



Exemplo

- Nó interno (que não é raiz nem folha):
 - Elementos: mínimo $\lceil N/2 \rceil - 1$ e Máximo $N-1$
 - Subárvores: mínimo $\lceil N/2 \rceil$ e Máximo N

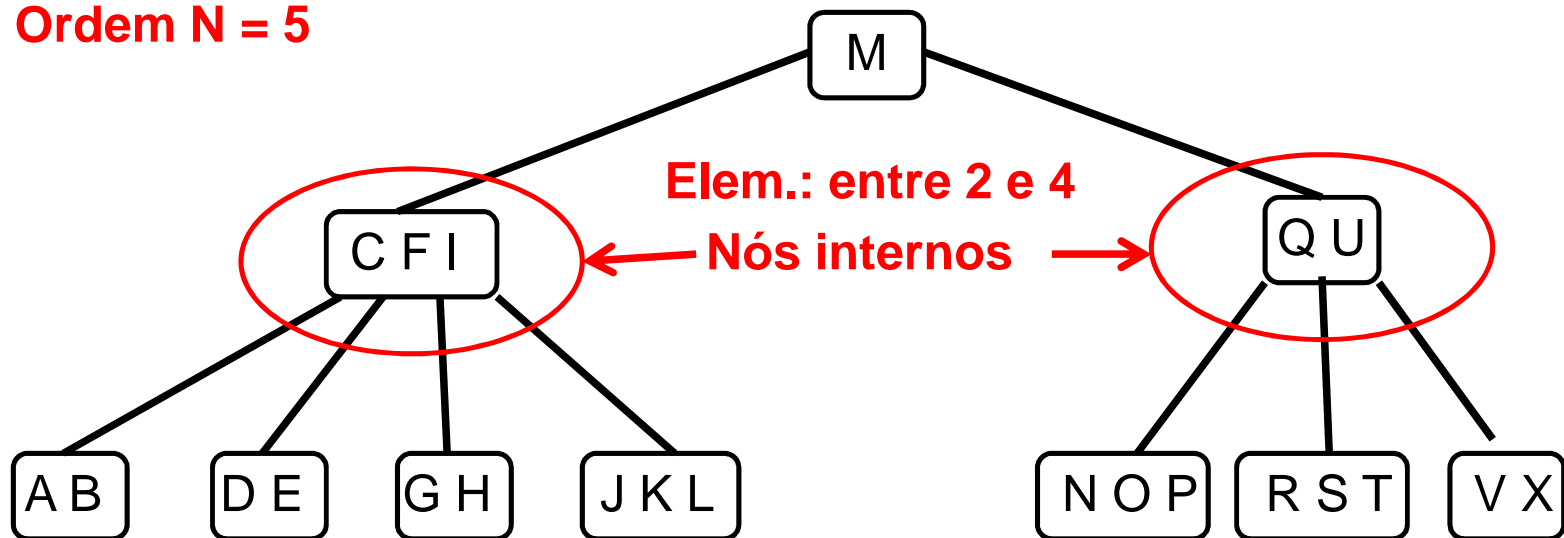
Ordem $N = 5$



Exemplo

- Nó interno (que não é raiz nem folha):
 - Elementos: mínimo $\lceil N/2 \rceil - 1$ e Máximo $N-1$
 - Subárvores: mínimo $\lceil N/2 \rceil$ e Máximo N

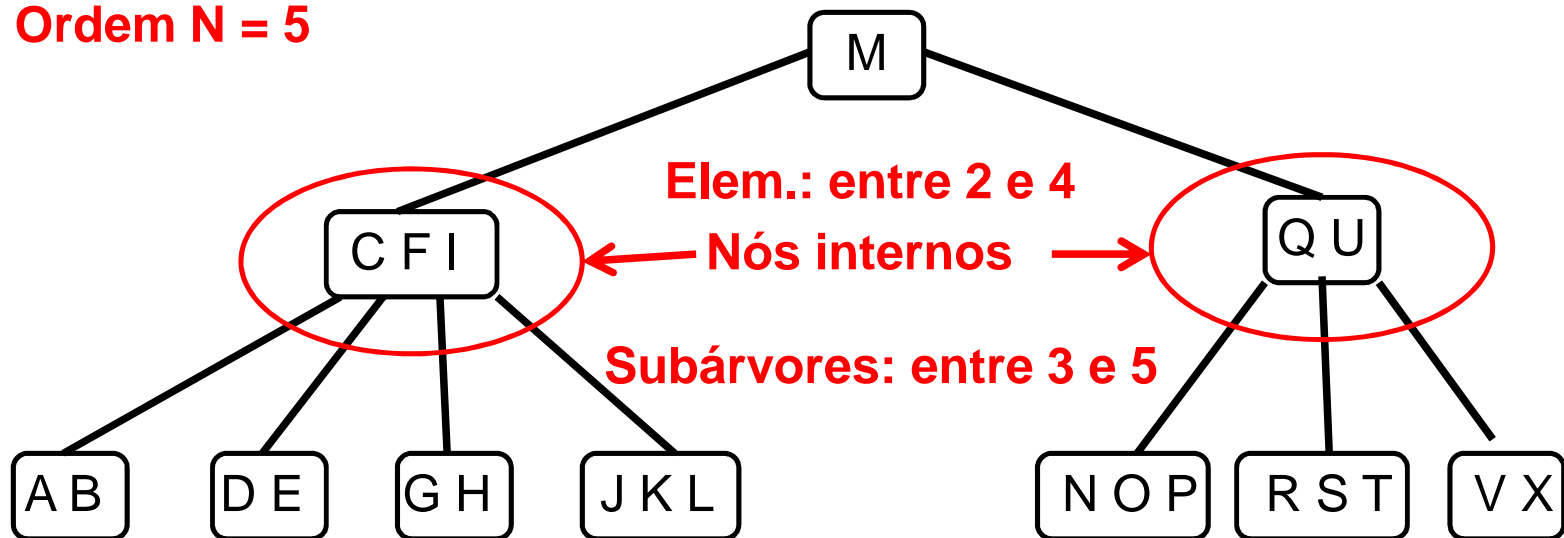
Ordem $N = 5$



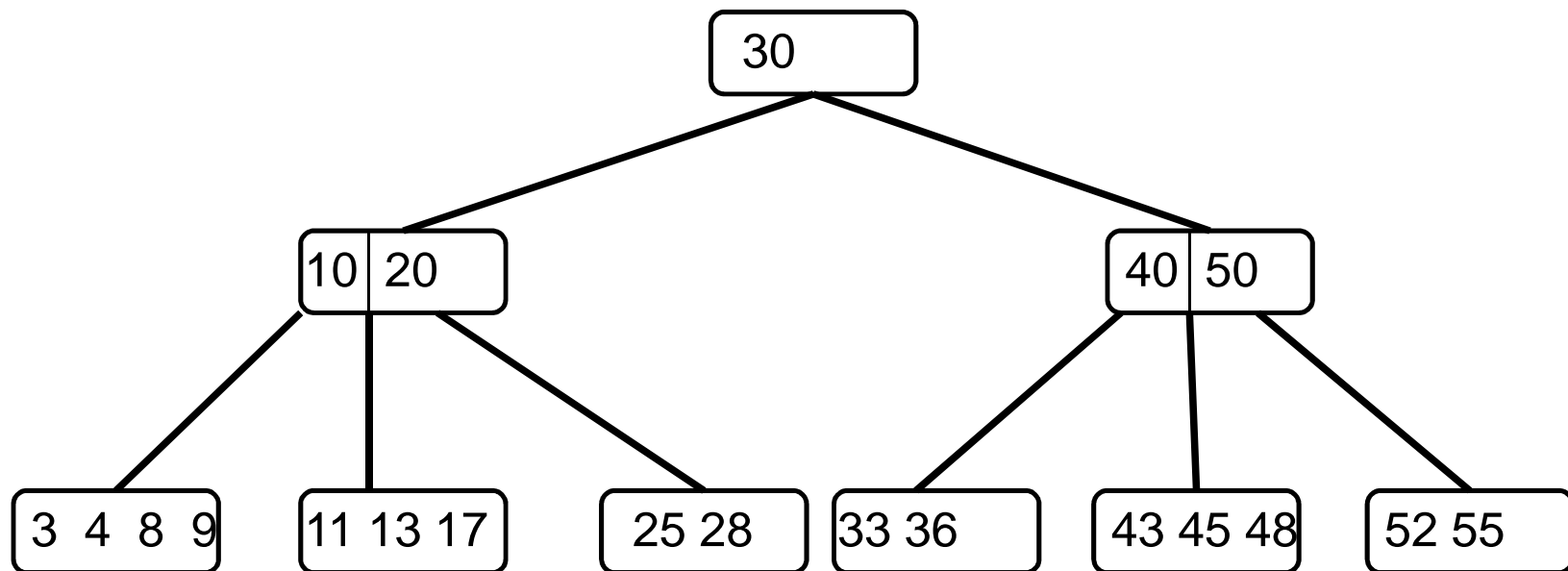
Exemplo

- Nó interno (que não é raiz nem folha):
 - Elementos: mínimo $\lceil N/2 \rceil - 1$ e Máximo $N-1$
 - Subárvores: mínimo $\lceil N/2 \rceil$ e Máximo N

Ordem $N = 5$

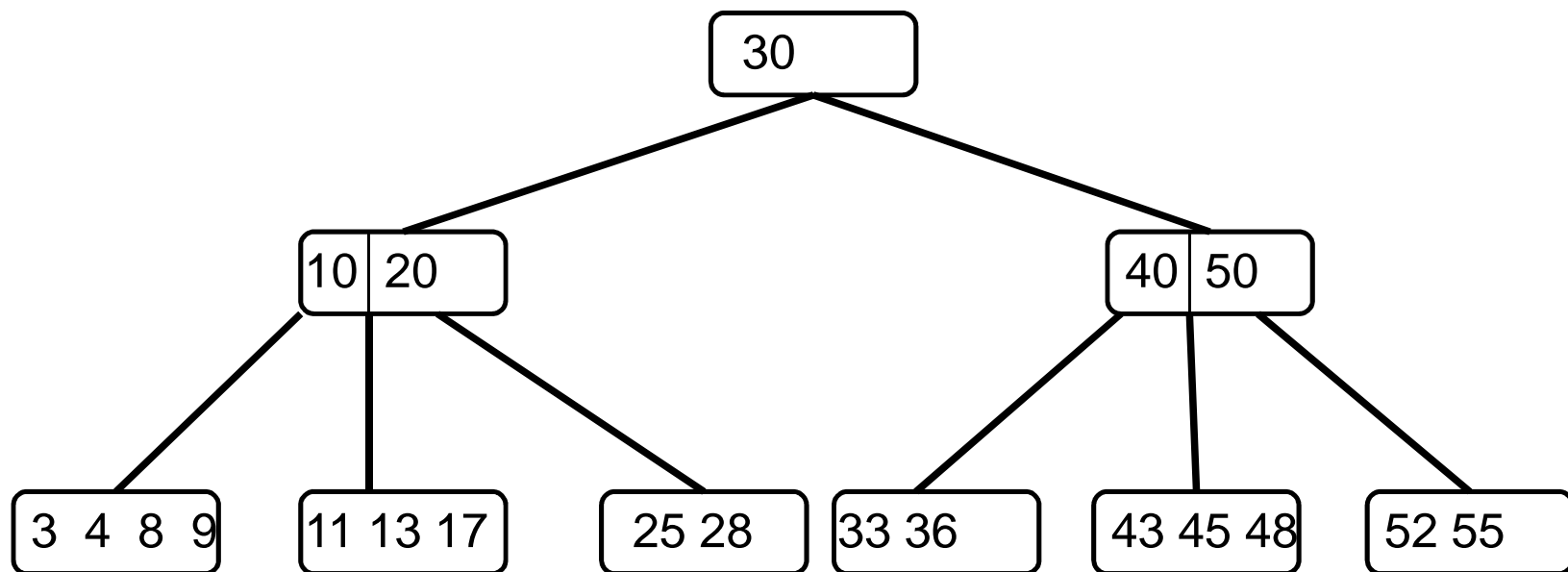


Exemplo de Árvore B



Exemplo de Árvore B

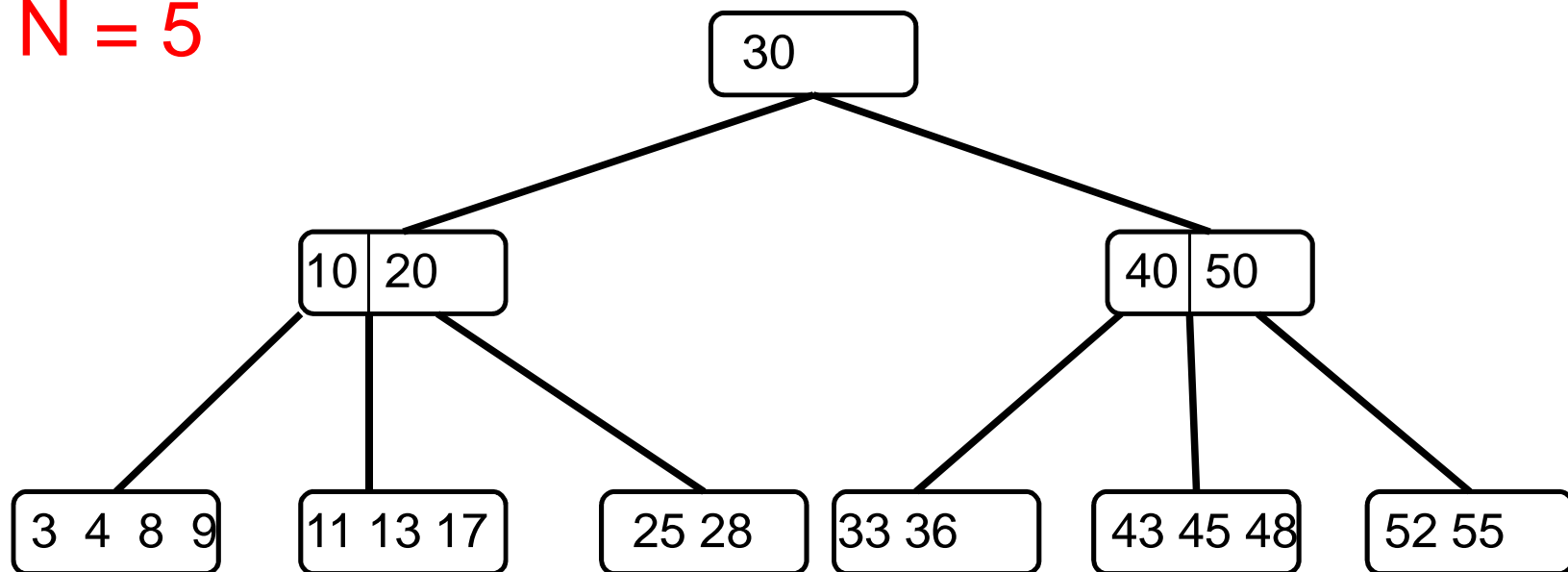
Qual a ordem da árvore ?



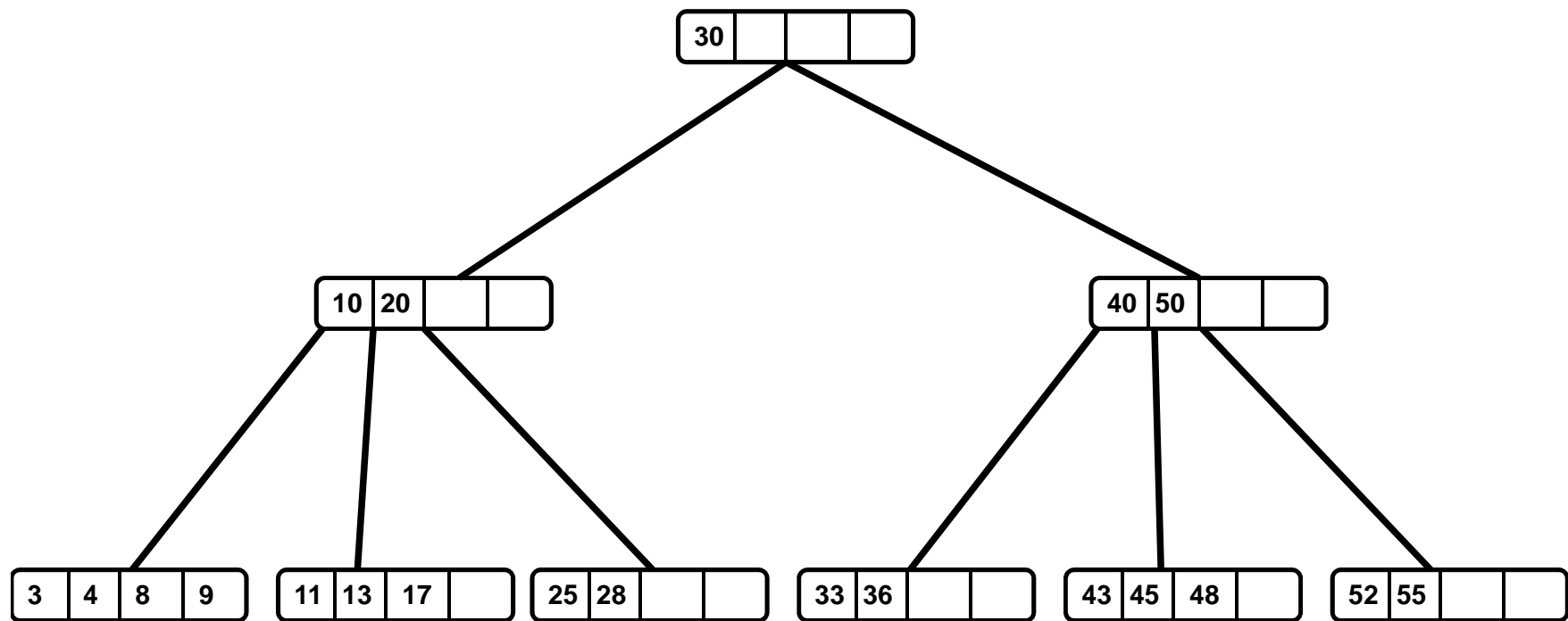
Exemplo de Árvore B

Qual a ordem da árvore ?

$N = 5$



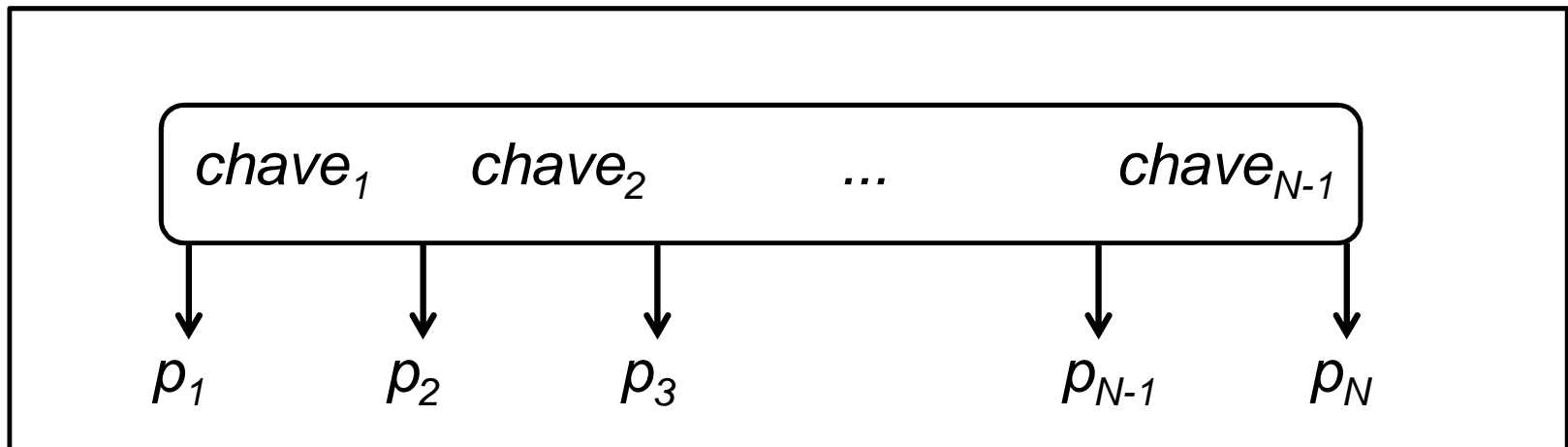
Exemplo de Árvore B de ordem 5



Estrutura de uma árvore B

```
#define N ? /* ordem da árvore B */  
struct no_arvoreB  
{  
    int num_chaves;  
    char chaves[N-1];  
    struct no_arvoreB *filhos[N];  
    bool folha;  
}
```

Ordem entre Elementos



Nó de uma árvore B de ordem N com N-1 elementos

Obs.: chave = elemento!

Operações em Árvore B

- Pesquisa
- Inserção
- Remoção

Pesquisa em Árvore B

- Semelhante a busca em árvores binárias de busca e AVL.


```

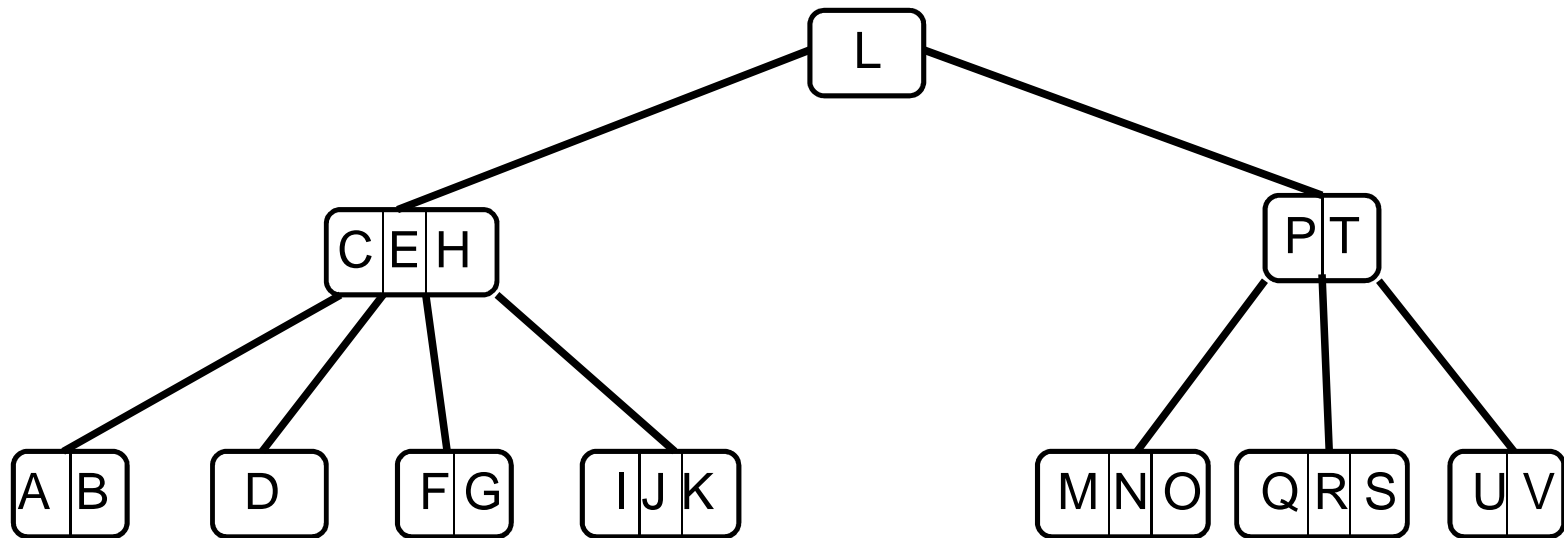
int busca_binaria(arvoreB *no, int info)
{
    int meio, i, f;  i = 0;
    f = no->num_chaves-1;

    while (i <= f)
    {
        meio = (i + f)/2;
        if (no->chaves[meio] == info)
            return(meio); //Encontrou. Retorna a posição em que a chave está.
        else if (no->chaves[meio] > info)
            f = meio - 1;
        else i = meio + 1;
    }
    return(i); //Não encontrou. Retorna a posição do ponteiro para o filho.
}

bool busca(arvoreB *raiz, int info)
{
    arvoreB *no;
    int pos; //posição retornada pelo busca binária.
    no = raiz;
    while (no != NULL)
    {
        pos = busca_binaria(no, info);
        if (pos < no->num_chaves && no->chaves[pos] == info)
            return(true);
        else no = no->filhos[pos];
    }
    return(false);
}

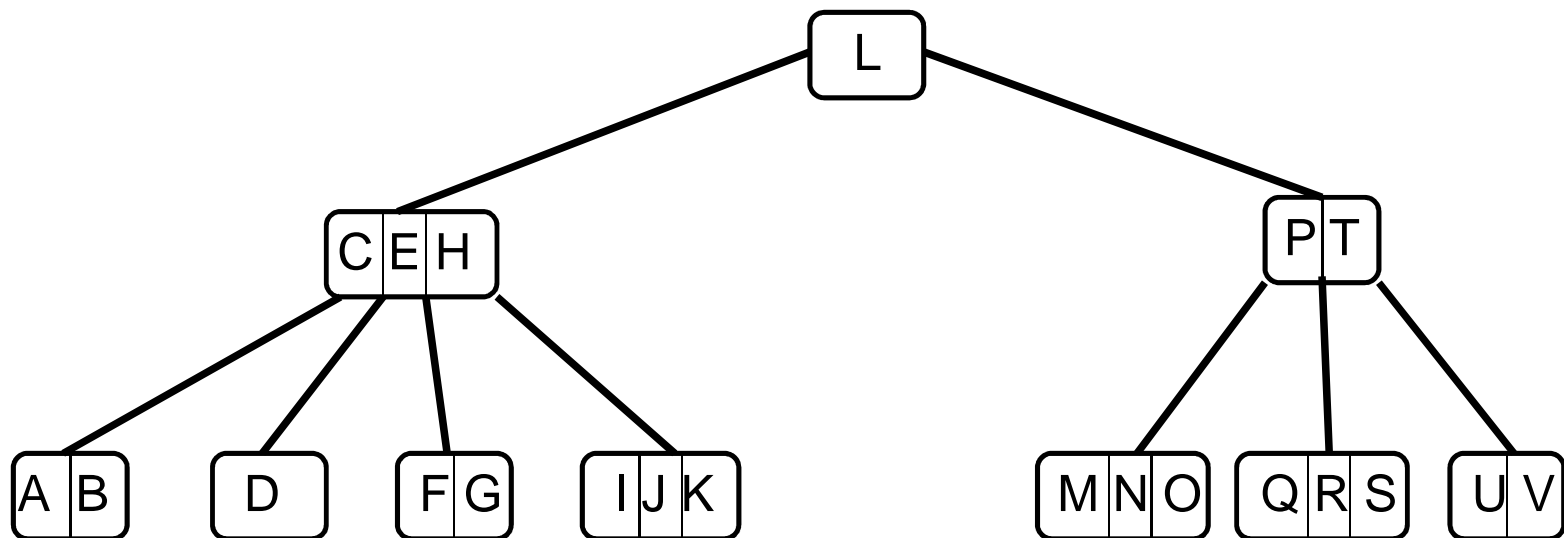
```

Pesquisa em Árvore B



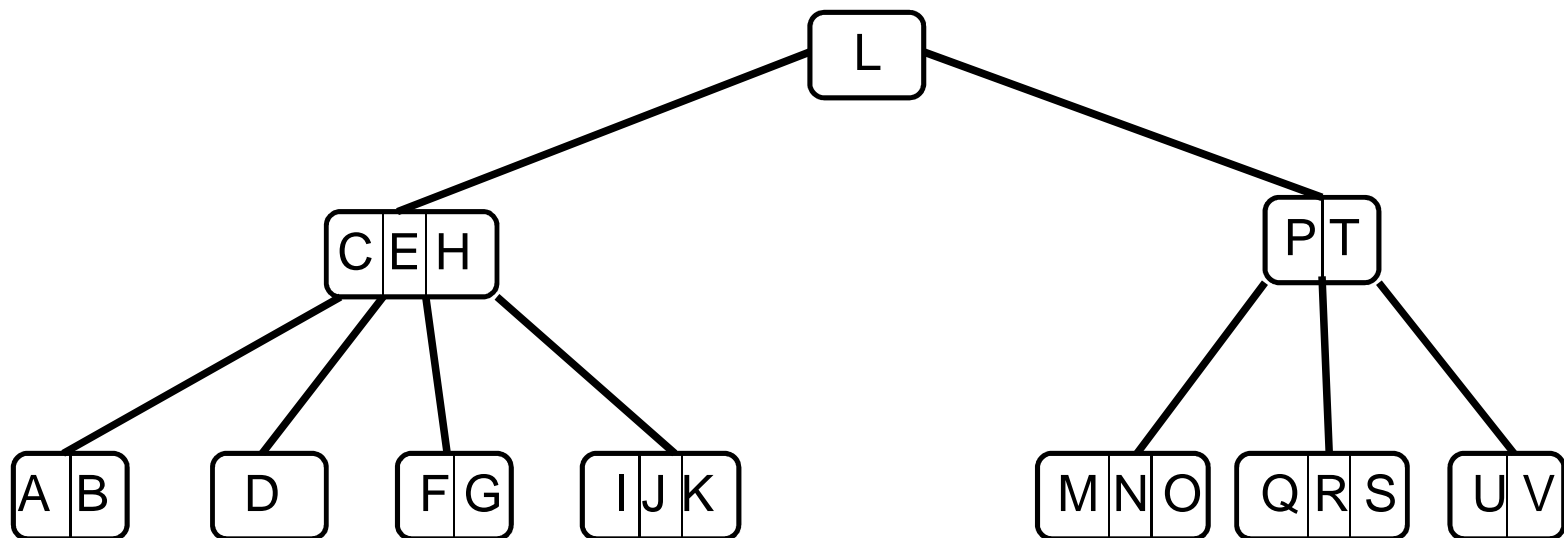
Pesquisa em Árvore B

Qual a ordem desta árvore?



Pesquisa em Árvore B

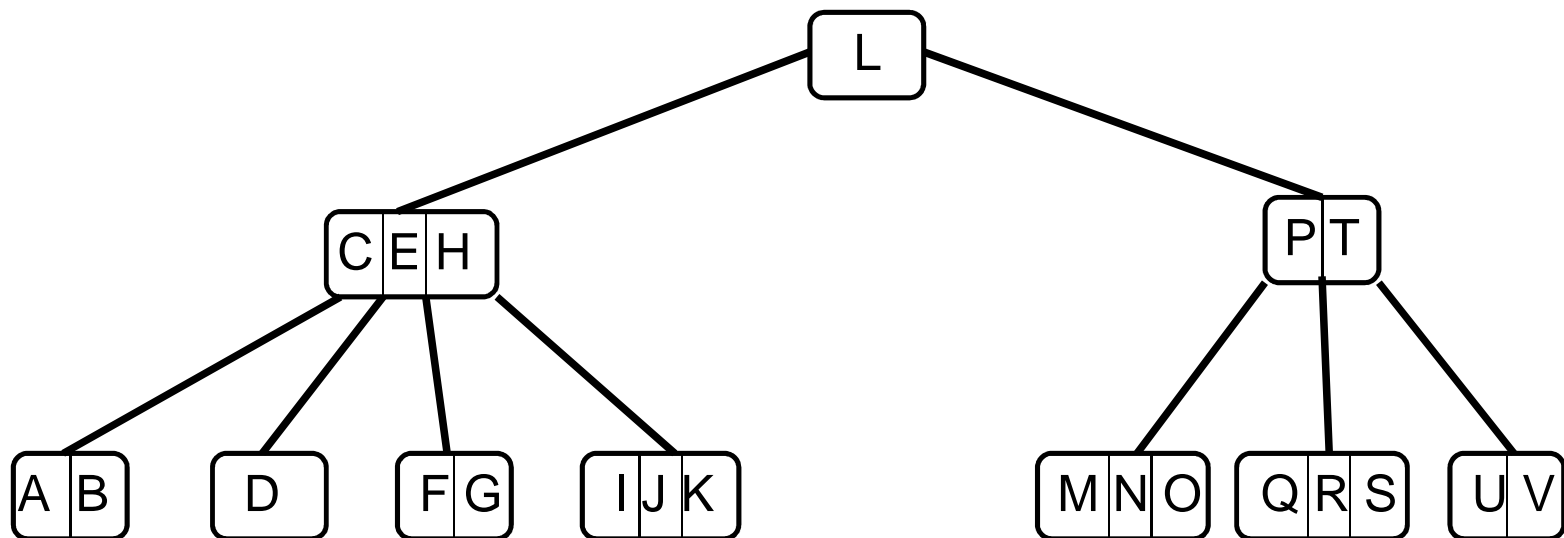
Qual a ordem desta árvore? $N = 4$



Pesquisa em Árvore B

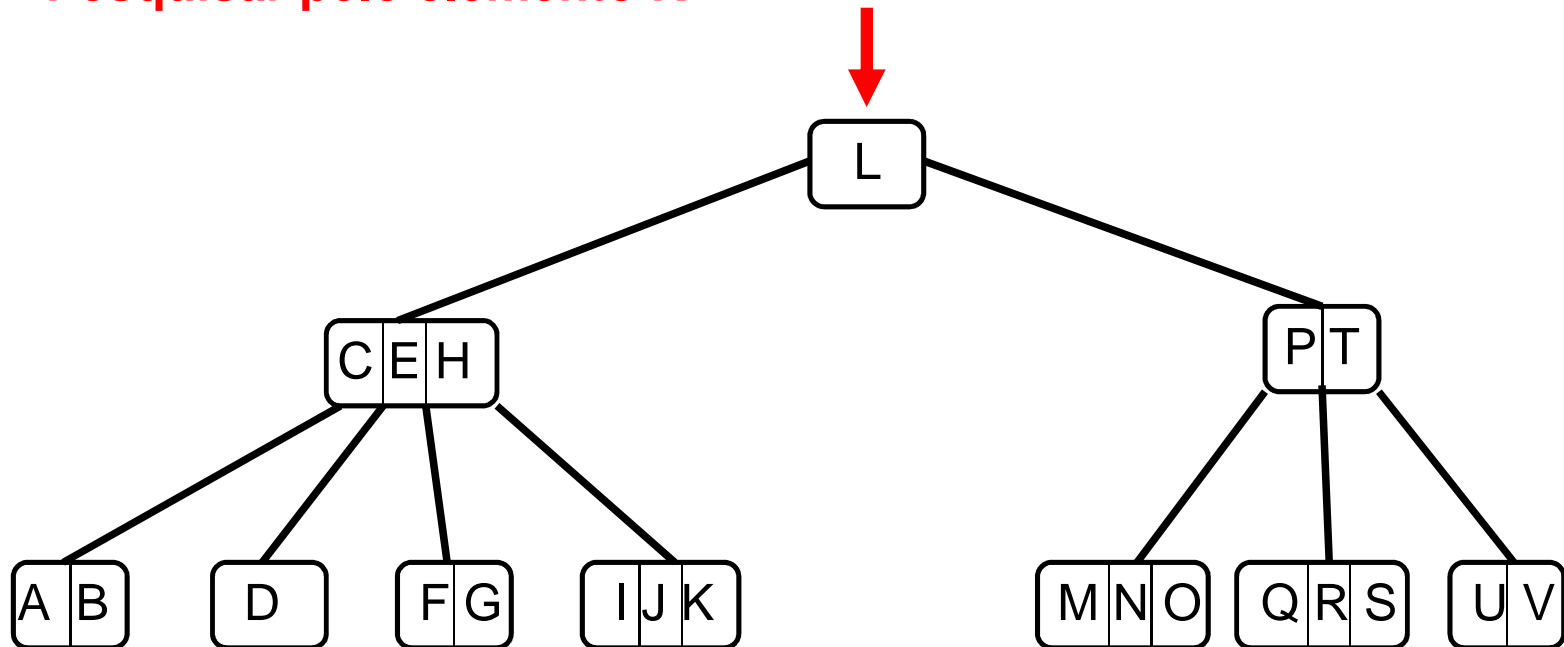
Qual a ordem desta árvore? $N = 4$

Pesquisar pelo elemento K



Pesquisa em Árvore B

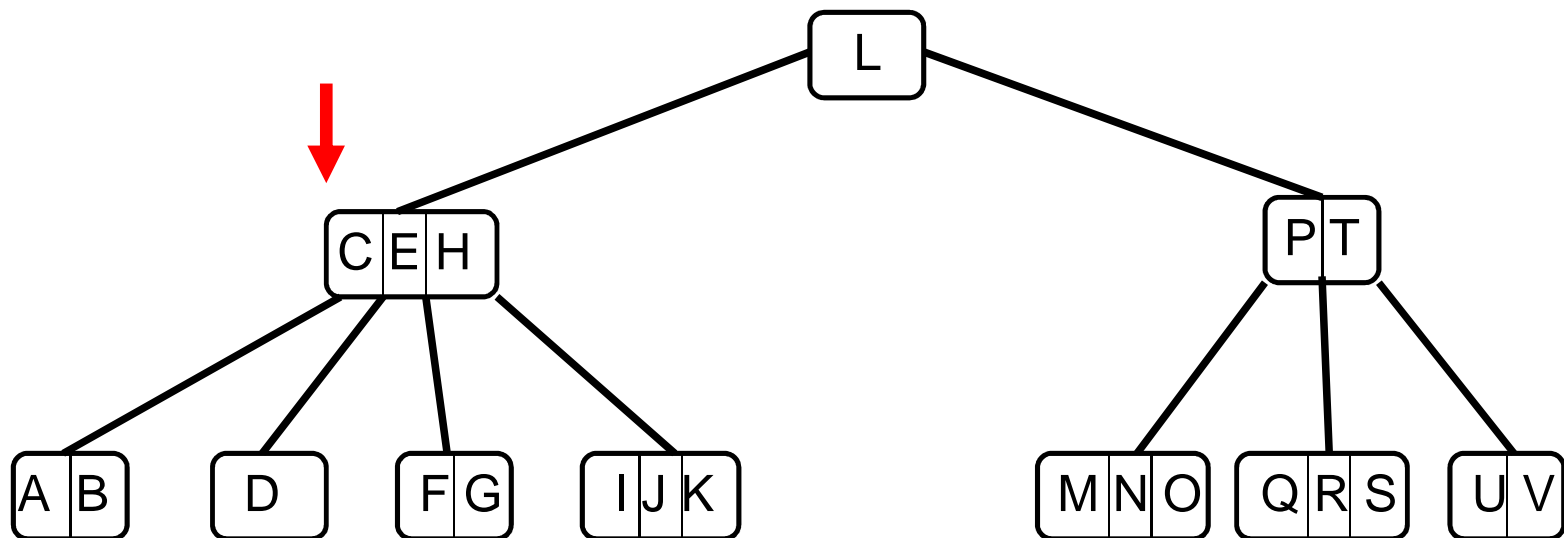
Qual a ordem desta árvore? $N = 4$
Pesquisar pelo elemento K



Pesquisa em Árvore B

Qual a ordem desta árvore? $N = 4$

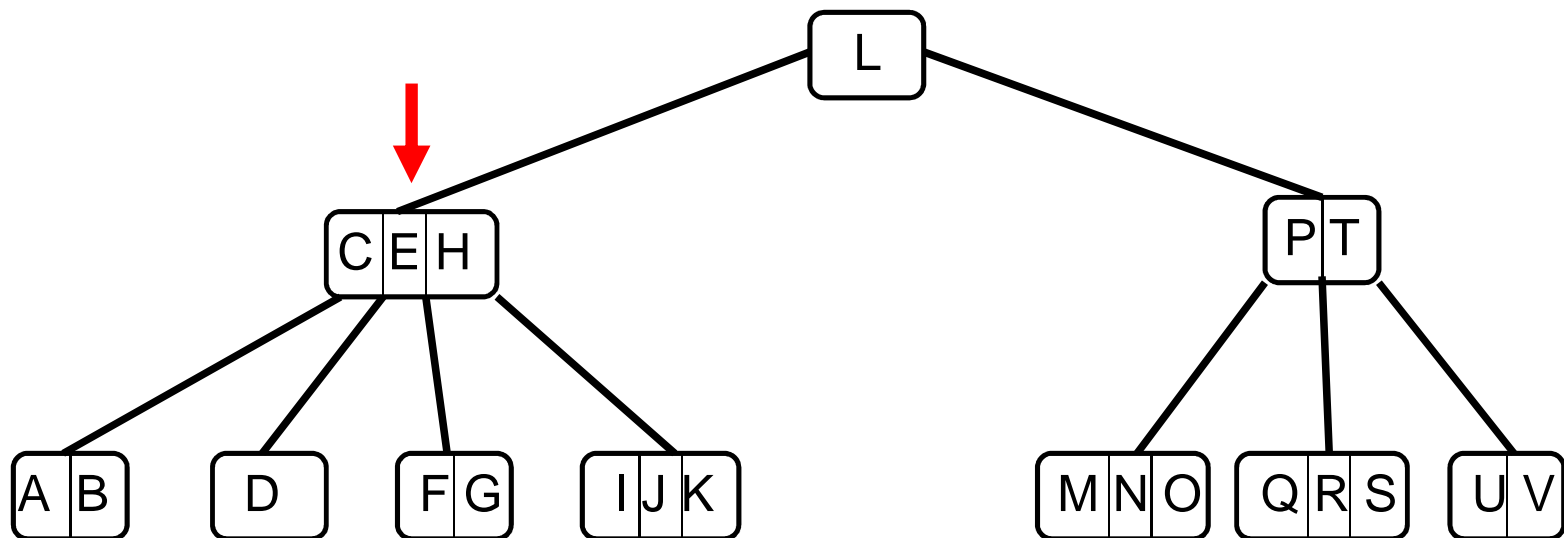
Pesquisar pelo elemento K



Pesquisa em Árvore B

Qual a ordem desta árvore? $N = 4$

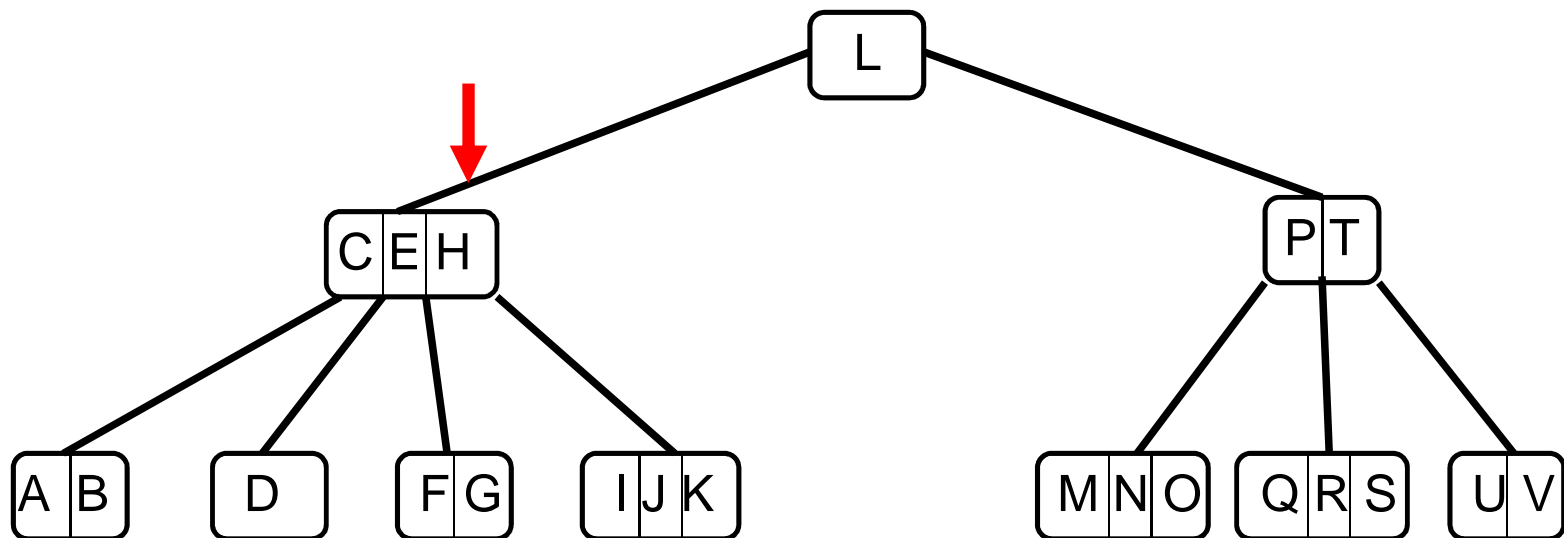
Pesquisar pelo elemento K



Pesquisa em Árvore B

Qual a ordem desta árvore? $N = 4$

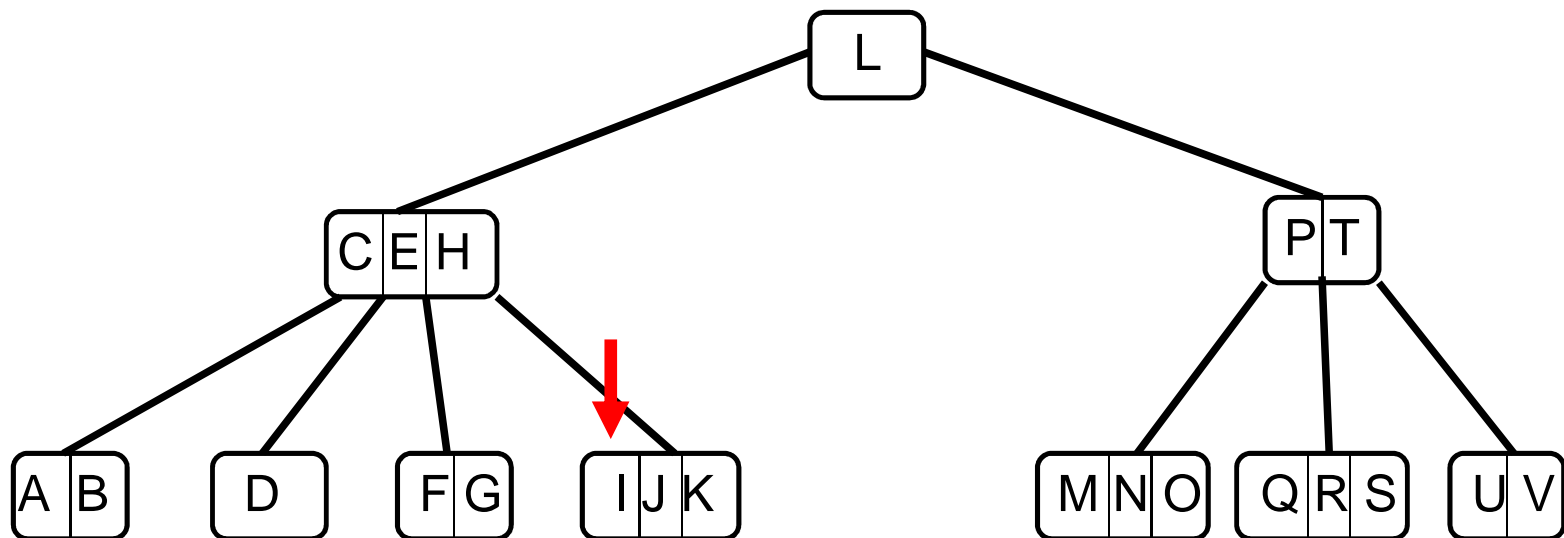
Pesquisar pelo elemento K



Pesquisa em Árvore B

Qual a ordem desta árvore? $N = 4$

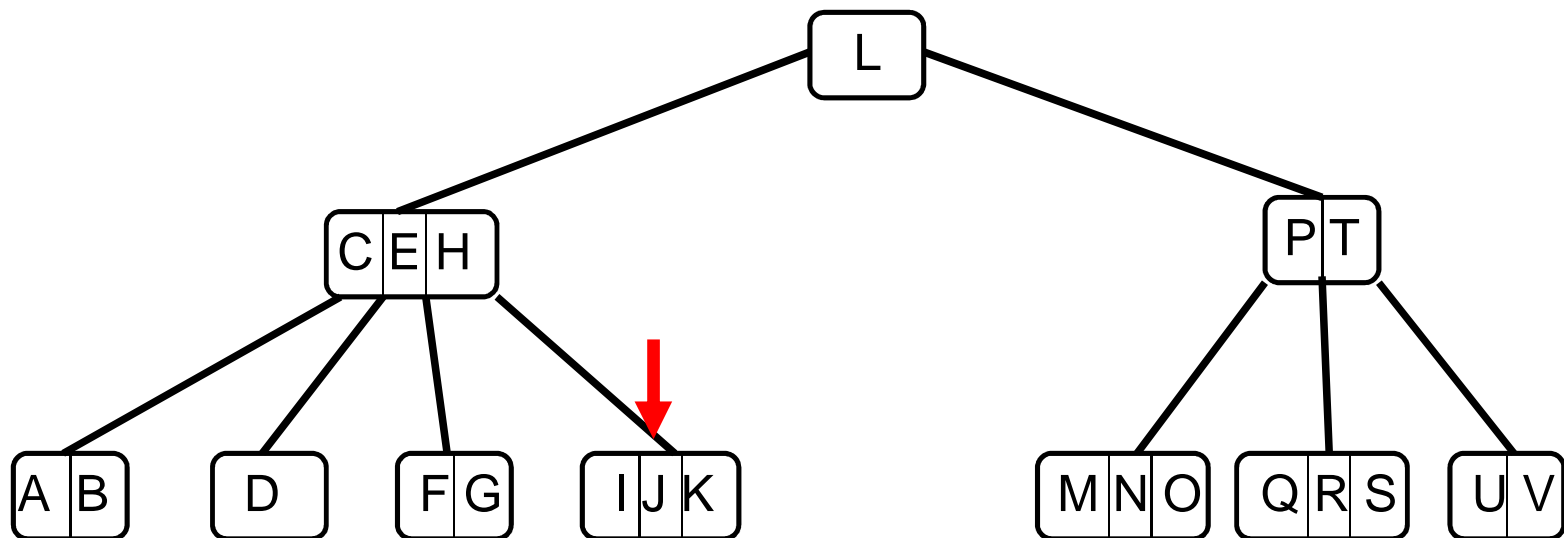
Pesquisar pelo elemento K



Pesquisa em Árvore B

Qual a ordem desta árvore? $N = 4$

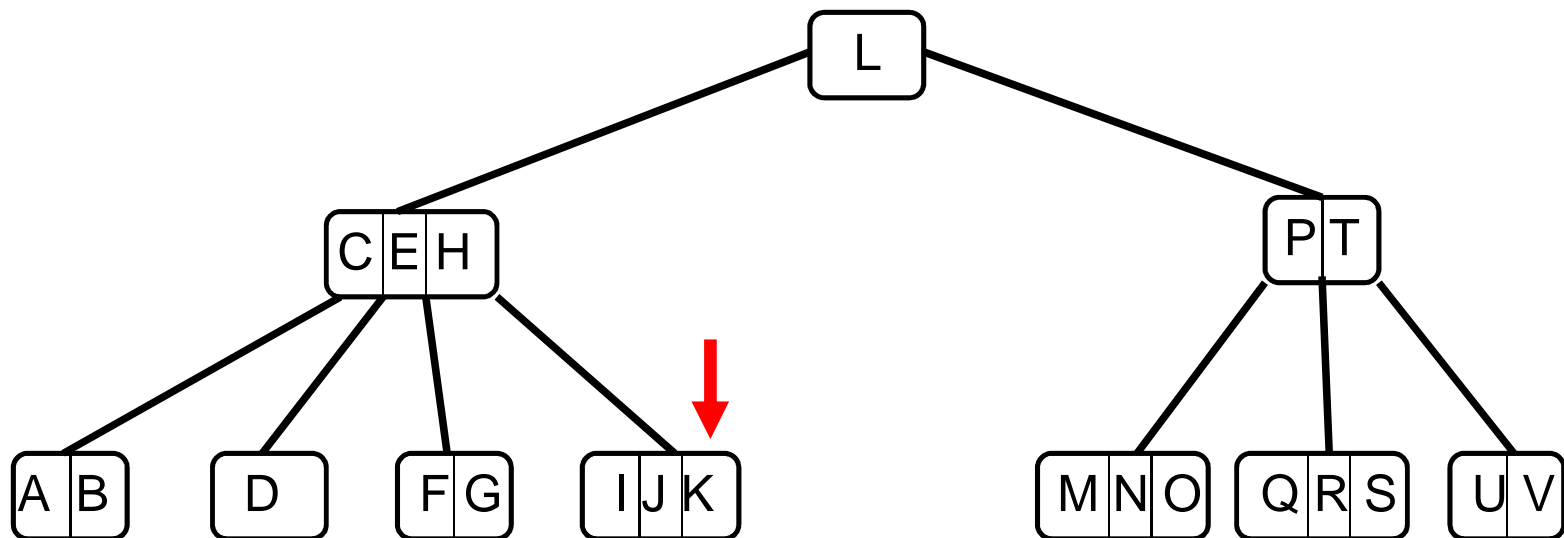
Pesquisar pelo elemento K



Pesquisa em Árvore B

Qual a ordem desta árvore? $N = 4$

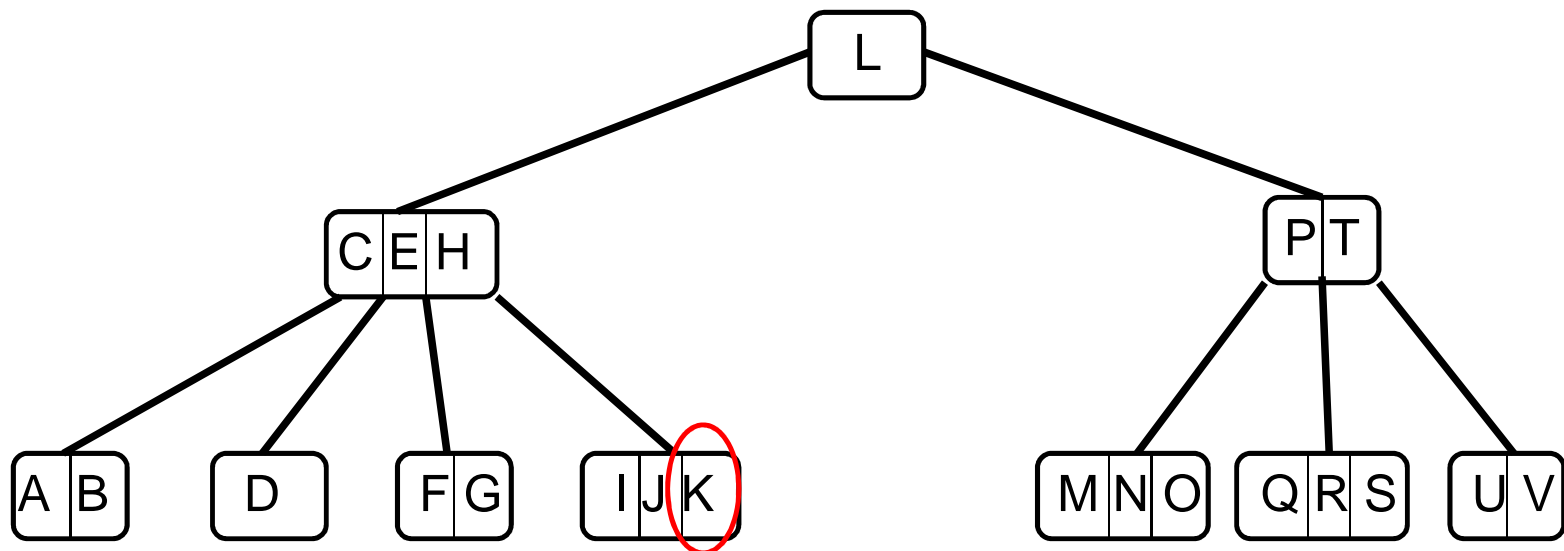
Pesquisar pelo elemento K



Pesquisa em Árvore B

Qual a ordem desta árvore? $N = 4$

Pesquisar pelo elemento K



Inserção

- Primeiro é preciso localizar a página onde o elemento deve ser inserido;
- Se a página onde o elemento deve ser inserido tiver menos de N elementos este é alocado nesta página;
- Se a página já estiver cheia o processo irá provocar a criação de uma nova página.

Inserção

Criação de nova página

- Primeiramente escolhe-se um valor intermediário na sequência ordenada de chaves da página incluindo-se a nova chave que deveria ser inserida
- Cria-se uma nova página e os valores maiores do que a chave intermediária são armazenados nessa nova página e os menores continuam na página anterior (operação de split).
- Esta chave intermediária escolhida deverá ser inserida na página pai, na qual poderá também sofrer *overflow* ou deverá ser criada caso em que é criada uma nova página raiz. Esta série de *overflows* pode se propagar para toda a árvore B, o que garante o seu **balanceamento** na inserção de chaves.

Inserção

- Exemplo: Inserir os valores 10, 20, 30, 40 e 50 em uma árvore B de ordem 5

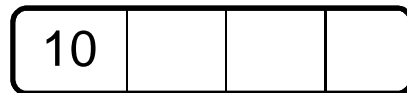
Inserção

- Exemplo: Inserir os valores 10,20,30,40 e 50 em uma árvore B de ordem 5



Inserção

- Exemplo: Inserir os valores 10,20,30,40 e 50 em uma árvore B de ordem 5



Inserção

- Exemplo: Inserir os valores 10,20,30,40 e 50 em uma árvore B de ordem 5

10	20		
----	----	--	--

Inserção

- Exemplo: Inserir os valores 10,20,30,40 e 50 em uma árvore B de ordem 5

10	20	30	
----	----	----	--

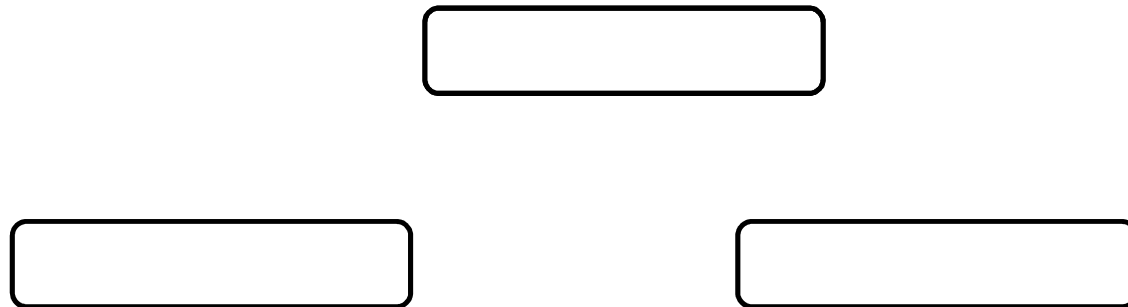
Inserção

- Exemplo: Inserir os valores 10,20,30,40 e 50 em uma árvore B de ordem 5

10	20	30	40
----	----	----	----

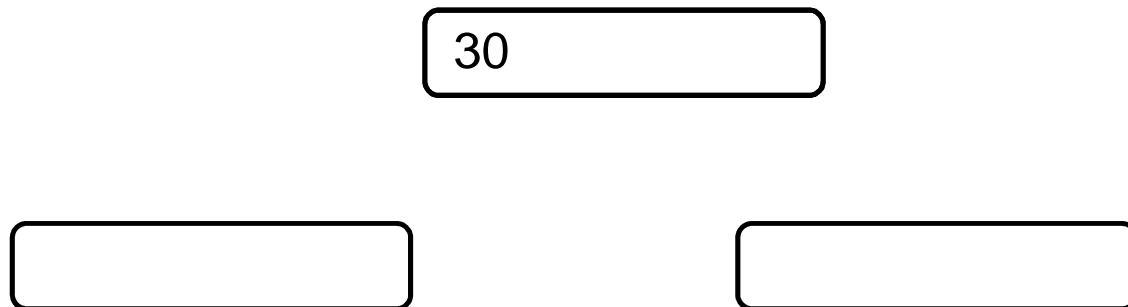
Inserção

- Exemplo: Inserir os valores 10,20,30,40 e 50 em uma árvore B de ordem 5



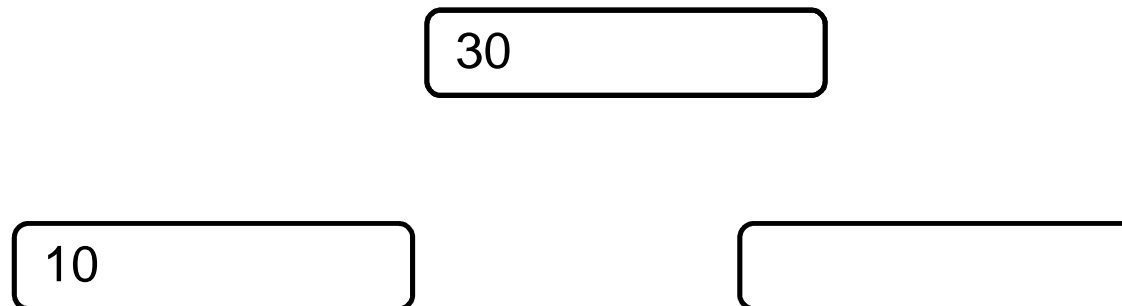
Inserção

- Exemplo: Inserir os valores 10,20,30,40 e 50 em uma árvore B de ordem 5



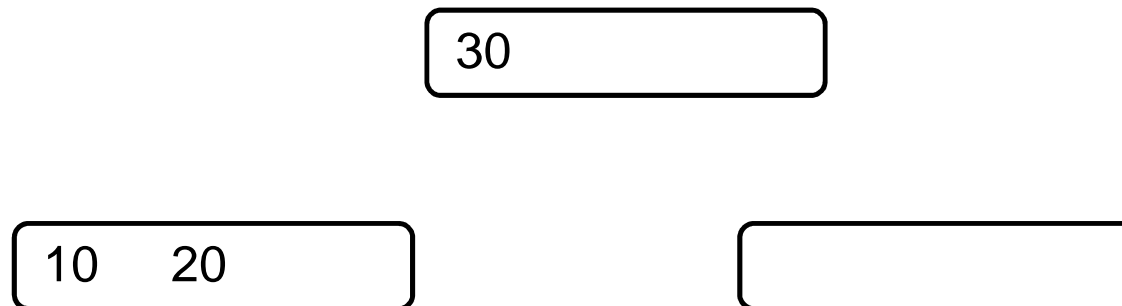
Inserção

- Exemplo: Inserir os valores 10,20,30,40 e 50 em uma árvore B de ordem 5



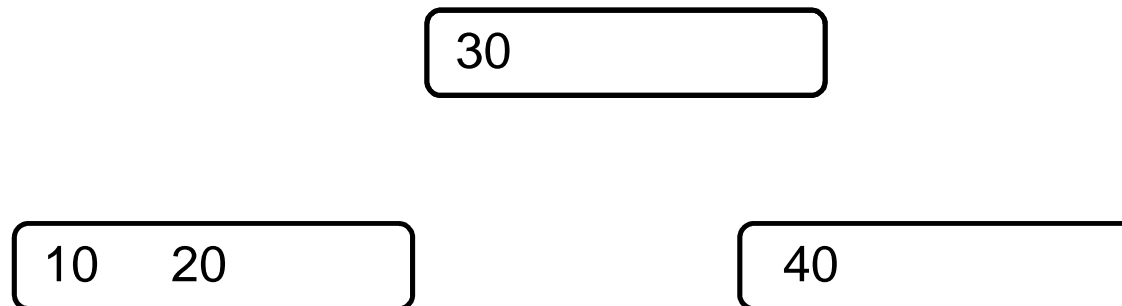
Inserção

- Exemplo: Inserir os valores 10,20,30,40 e 50 em uma árvore B de ordem 5



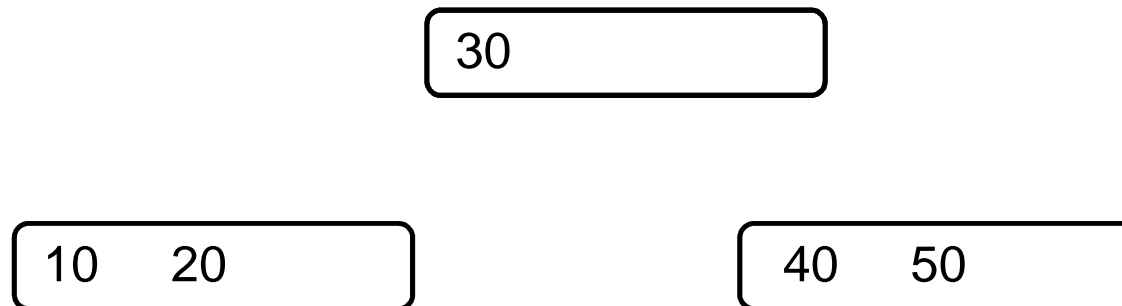
Inserção

- Exemplo: Inserir os valores 10,20,30,40 e 50 em uma árvore B de ordem 5



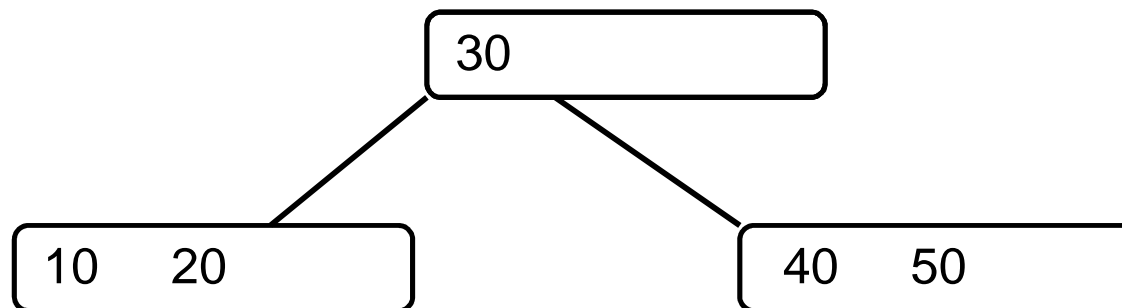
Inserção

- Exemplo: Inserir os valores 10,20,30,40 e 50 em uma árvore B de ordem 5



Inserção

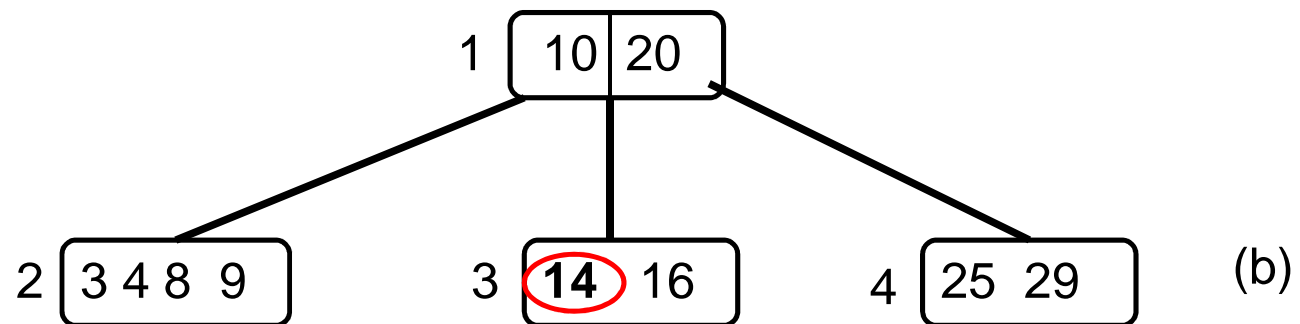
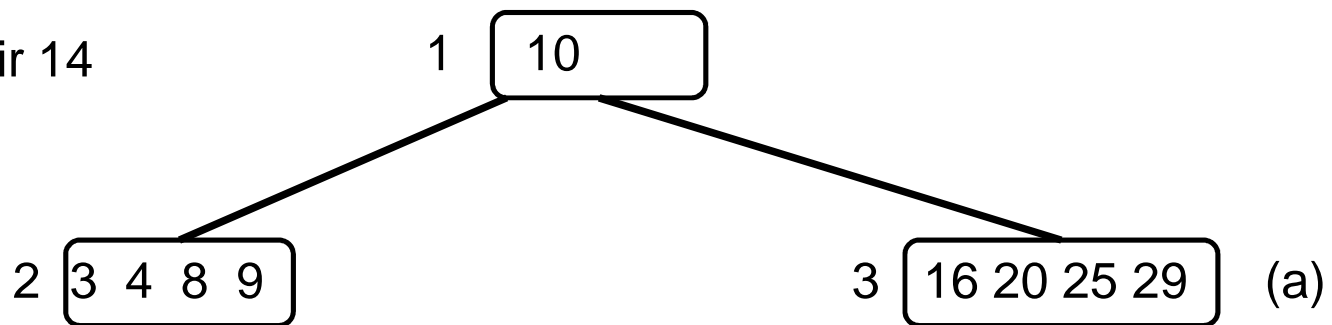
- Exemplo: Inserir os valores 10,20,30,40 e 50 em uma árvore B de ordem 5



Inserção em uma árvore B

- Ordem 5

Inserir 14



Inserção em uma árvore B de ordem 5

- 1. O registro contendo a chave 14 não é encontrado na árvore, e a página 3 (onde o registro contendo a chave 14 deve ser inserido) está cheia;
- 2. A página 3 é dividida em duas páginas, o que significa que uma nova página 4 é criada.
- 3. Os N registros (no caso são cinco registros) são distribuídos igualmente entre as páginas 3 e 4, e o registro do meio (no caso o registro contendo a chave 20) é movido para a página pai no nível acima.

Inserção

- Neste esquema de inserção, a página pai tem de acomodar um novo elemento. Se a página pai também estiver cheia, então o mesmo processo de divisão tem de ser aplicado de novo.
- No pior caso, o processo de divisão pode propagar-se até a raiz da árvore, e, neste caso, ela aumenta sua altura e um nível.
- É interessante observar que uma árvore B somente aumenta sua altura com a divisão da raiz.

Referências

- Projeto de Algoritmos – Nívio Ziviani
- Estruturas de Dados usando C – Tenenbaum
- Algoritmos – Teoria e Prática – Cormen
- B-Trees - animação
 - <https://www.cs.usfca.edu/~galles/visualization/BTree.html>

Exercício

1. Construa as seguintes árvores:

a) Ordem 3, valores: 5, 7, 14, 30, 21

b) Na a árvore obtida em a) insira: 35, 37