Árvores AVL e Árvores B

Jeane Melo

Roteiro

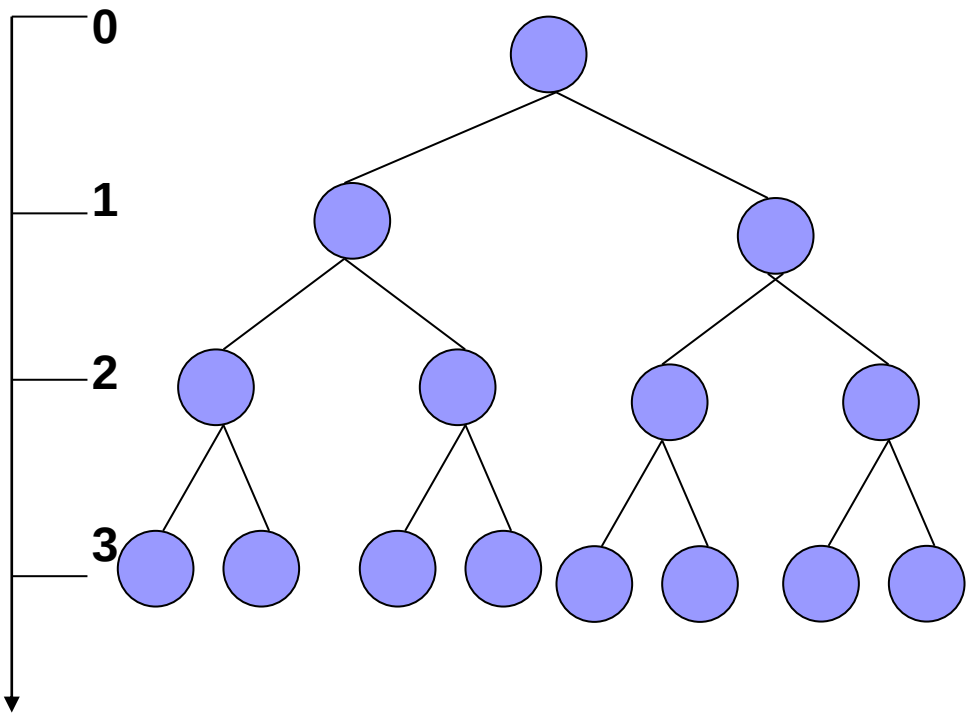
- Árvores Binárias
- Árvores AVL
  - Definição
  - Motivação
  - Balanceamento
  - Operações de rebalanceamento
- Árvores B
  - Introdução

Árvores Binárias

- Árvores binárias
  - Cada nó tem no máximo duas sub-árvores
  - Quando há apenas uma sub-árvore indicar se é esquerda ou direita
- Árvore binária completa
  - Cada nó ou é uma folha ou tem grau exatamente igual a dois
  - Não existe nenhum nó de grau 1

Árvores Binárias

- Se uma árvore binária tem m nós no nível d então ela terá no máximo 2m nós no nível d+1
- Observe que o nível i de uma árvore binária poderá conter no máximo 2^i nós



Árvores AVL

- Proposta pelos matemáticos russos Georgii Adelson Vel'sky e Yevgeniy Landis: "Algoritmos para organização de informação" [1962]

- Árvore de busca binária auto-balanceada
  - As alturas das sub-árvores esquerda e direita diferem por no máximo 1
  - Complexidade da ordem de O(logn), onde n é o número de nós, para operações de inserção, remoção e busca por elementos.

Árvores AVL - Definição

- Uma árvore binária T é denominada AVL quando, para todo nó v, a altura das 2 subárvores, esquerda e direita, satisfazem

$$fb(v) = | \text{altura(dir)} - \text{altura(esq)} | \leq 1$$

onde fb(v) é o fator de balanceamento do nó v.

Árvores AVL - Motivação

- As operações básicas sobre árvores de busca binária têm complexidade proporcionais à altura da árvore
- No caso de uma árvore binária completa – O(logn)
- Se a árvore é uma cadeia linear de n nós as mesma operações custa O(n) no pior caso.

Árvores AVL

- A altura de uma ALV é proporcional a log n, assim, o custo para inserir e retirar elementos é da O(log n)

- Após operações de inserção e remoção verifica-se o fator de balanceamento para manter esta propriedade da árvore.

Balanceamento

- “Uma árvore AVL é dita balanceada quando a diferença entre as alturas das sub-árvores não é maior do que um.”

- É possível demonstrar que a altura de uma árvore AVL é proporcional a $\log_2 n$ (~ 1.5* $\log_2 n$).

Árvores AVL

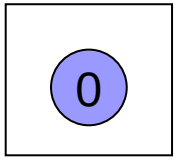
- As operações básicas sobre uma AVL são as mesmas utilizadas em árvores binárias de busca: Inserção, Remoção, Busca.
- Porém, após realizar uma inserção ou remoção verifica-se o fator de balanceamento. Se este difere de 1, 0 ou -1, se faz necessária uma operação de rotação para “rebalancear” a árvore.

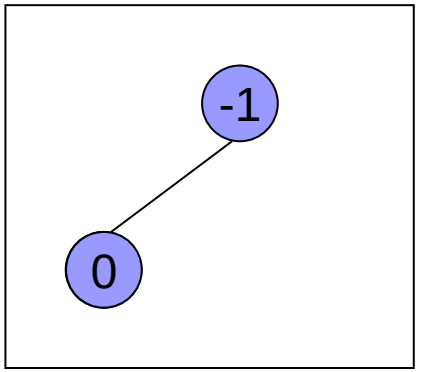
Árvores AVL

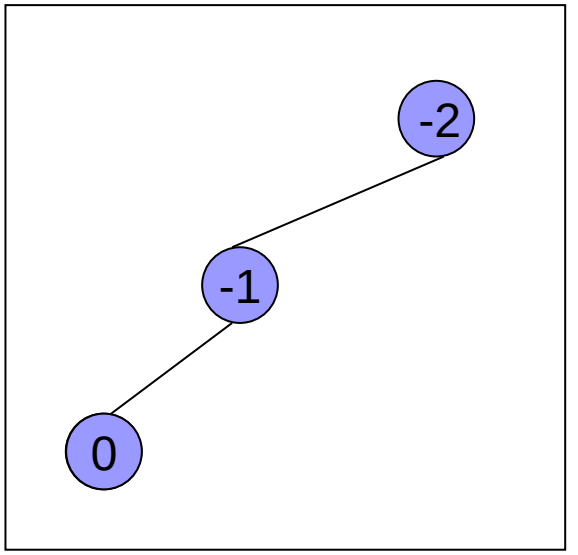
- Como calcular o fator de balanceamento?
  - Altura (sub-árvore da direita) – Altura (sub-árvore da esquerda)
- Como calcular a altura da árvore?

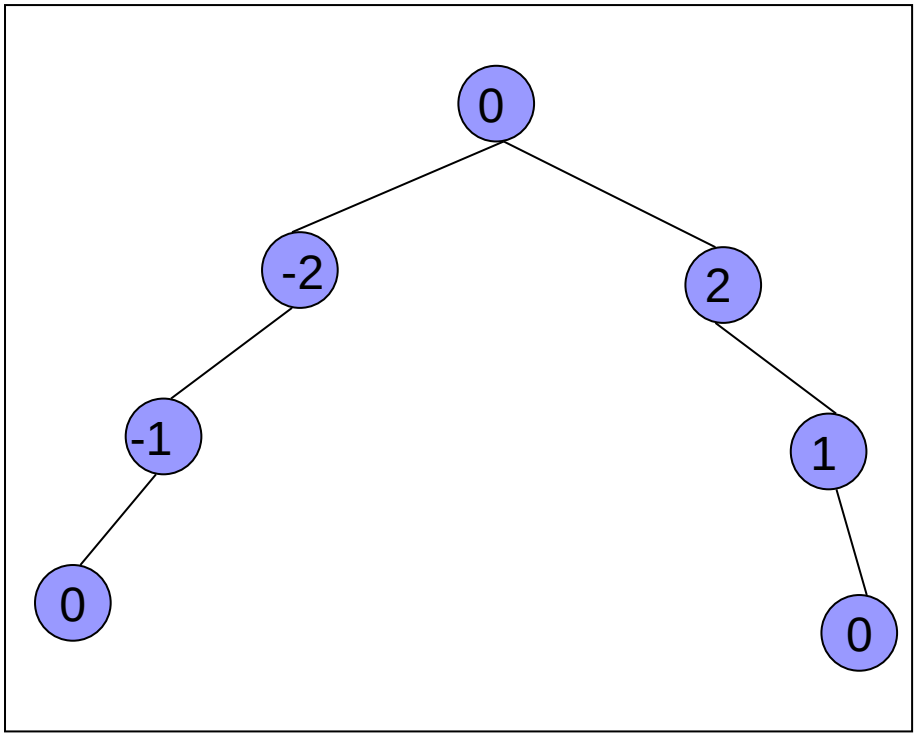
```
int altura (arvore r) {
  if (r == NULL) return -1;
  else {
        int he = altura (r->esq);
        int hd = altura (r->dir);
        if (he < hd)
            return hd + 1;
        else
            return he + 1;
         }
 }
```

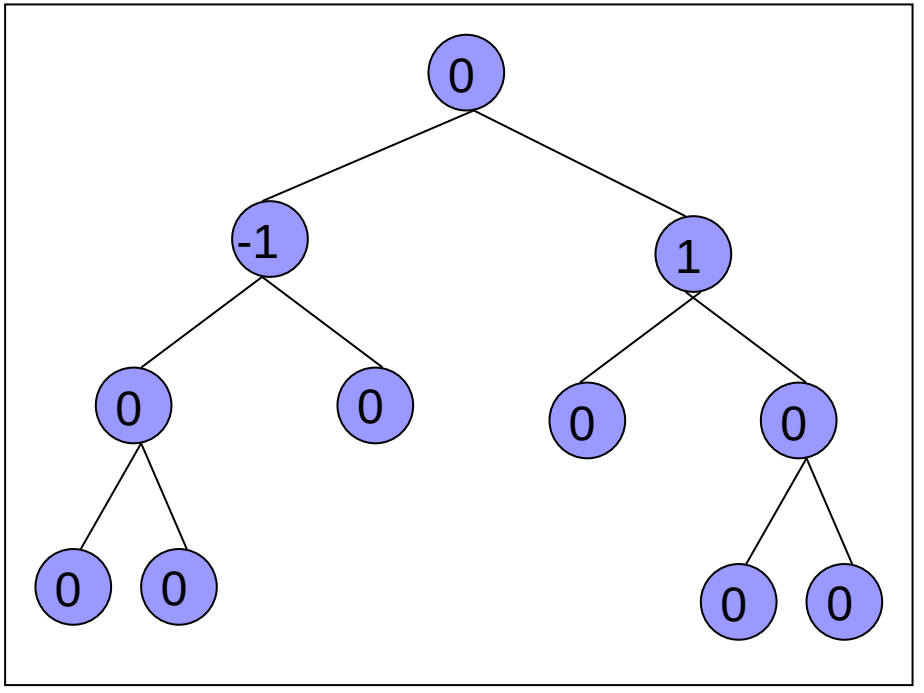
Exemplos











AVL - Rebalanceamento

- Operações de Rotação
  - Rotação à esquerda
  - Rotação à direita
  - Rotação dupla à esquerda
  - Rotação dupla à direita
- Usadas após inserção ou remoção de um elto.

Inserção em uma AVL

AVL-Inserir (T, x)
Entrada: Árvore AVL e um elto x para inserção em T.
Saída: Árvore AVL (T + x)
Início

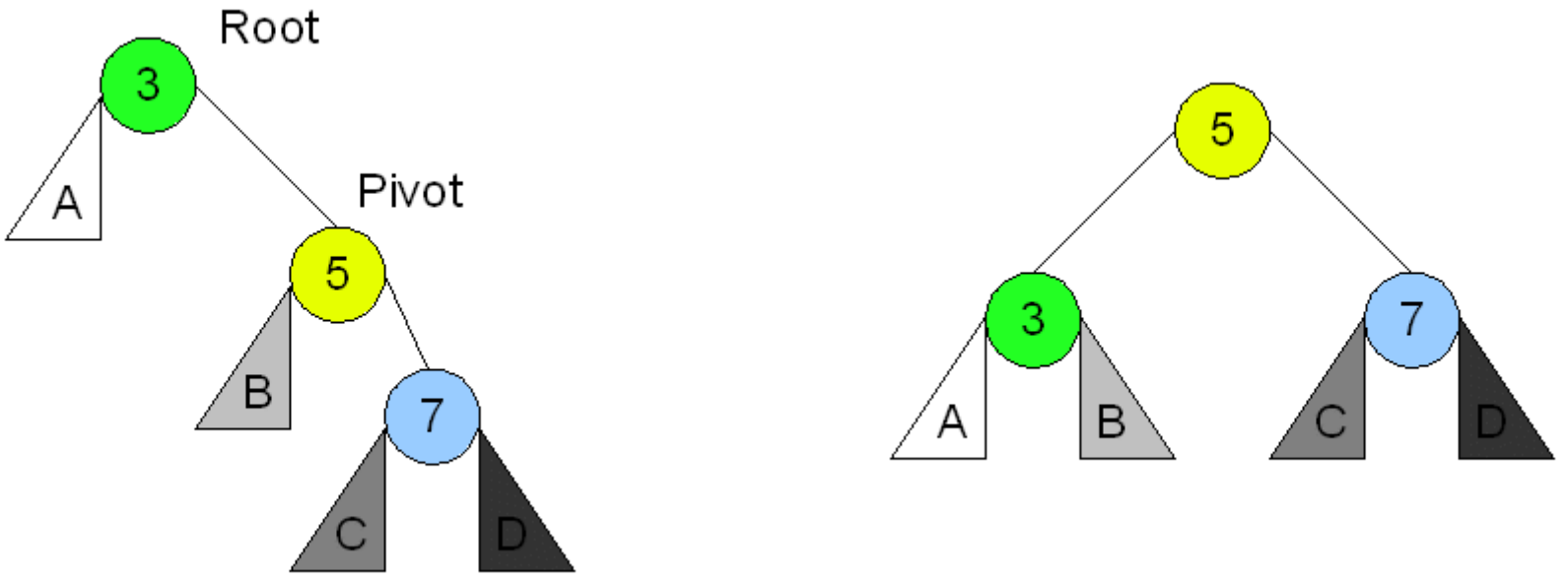
1. Use o algoritmo de inserção para árvore de busca binária.
2. Se (T + x) é AVL então devolva (T + x)
3. senão T' = AVL-Balance (T + x).
4. Devolva T' .

Fim

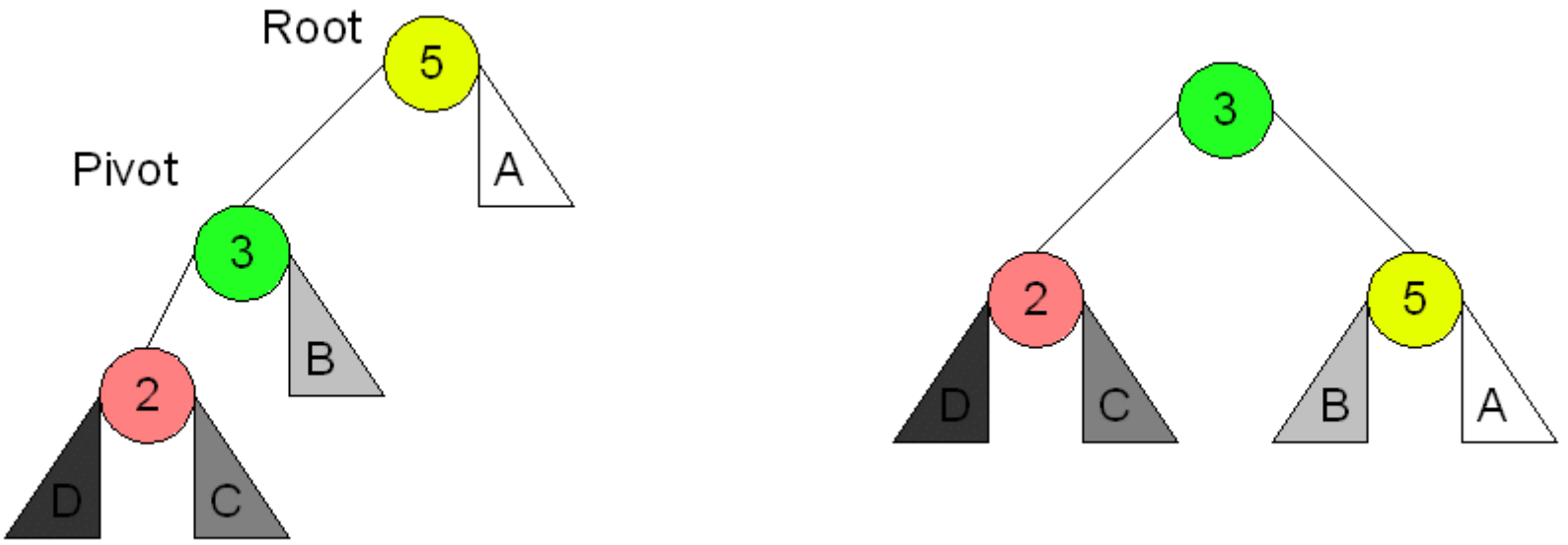
Remoção em uma AVL

AVL-Remoção(T,x)
Entrada: árvore AVL e o nó x a ser removido em T.
Saída: árvore AVL (T – x).
Início
1. Execute a remoção como na árvore binária de busca.
2. Verifique se a árvore ficou desregulada (use o fator de balanceamento como na inserção).
3. Execute uma ou mais rotações (simples ou dupla).
4. Devolva (T – x).
Fim

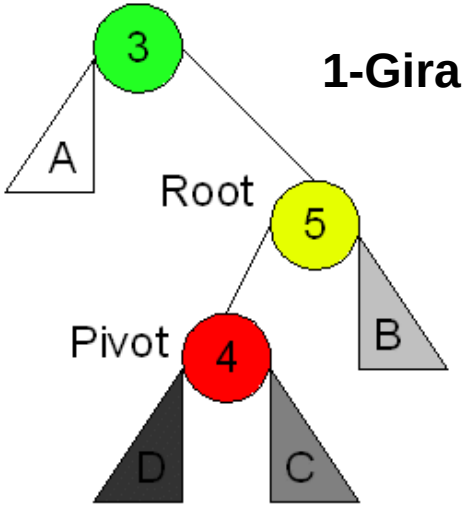
Rotação à esquerda



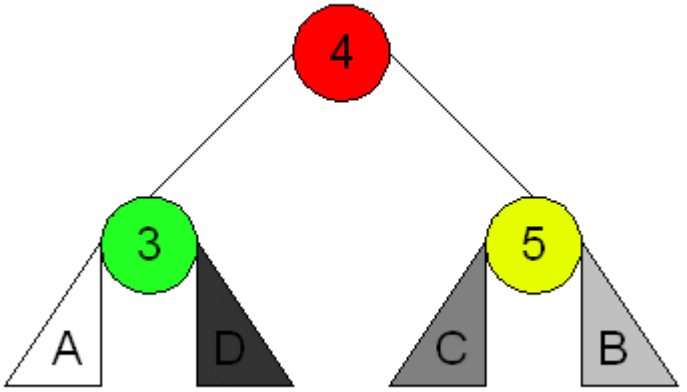
Rotação à direita



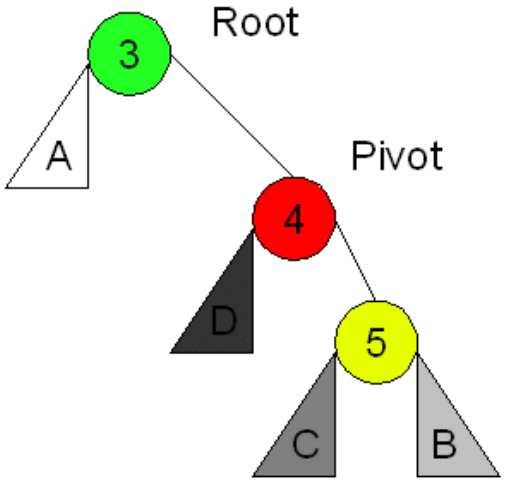
Rotação dupla à esquerda



1-Gira à direita

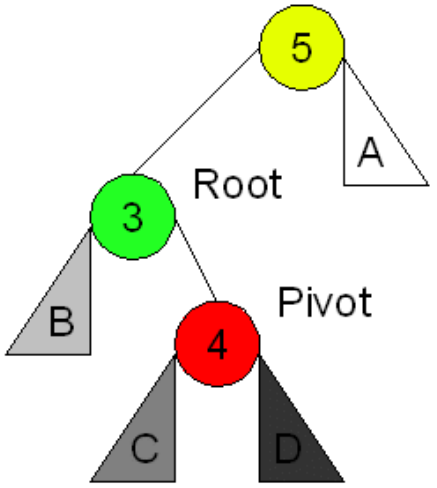


3-Árvore resultante

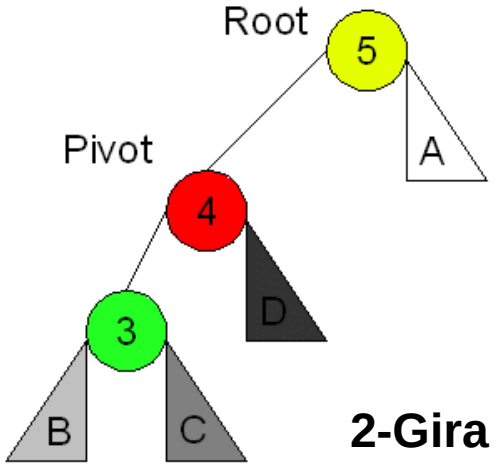


2-Gira à esquerda

Rotação dupla à direita

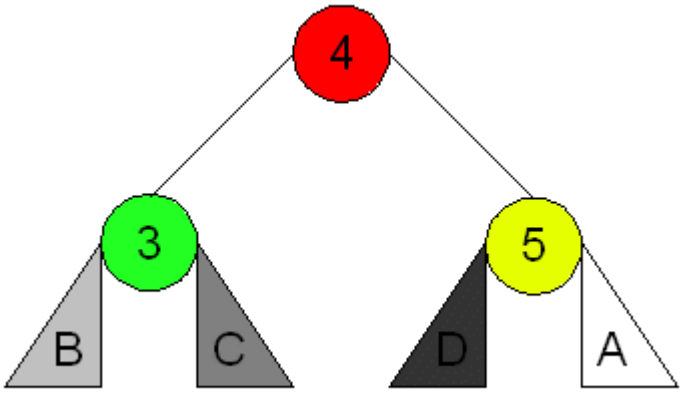


1-Gira à esquerda



2-Gira à direita

3-Árvore resultante



Árvores B

- As árvores de busca, quando balanceadas, permitem acesso em tempo logarítmico
- No caso das AVL o tempo é $\log_2 (n)$
- Podemos tentar melhorar esse tempo?

Árvores B

- Uma forma seria procurar fazer a busca em tempo $\log_k (n)$, com $k > 2$

Árvores B

Isto pode ser feito aumentando o número de elementos em cada nó.

Árvores B

- Árvores balanceadas
- Projetadas para trabalhar com dispositivos de armazenamento secundário (discos magnéticos)
- Otimizar as operações de entrada e saída nos dispositivos (minimizar o número de acessos ao disco)
- Um nó em uma árvores B pode ter muitos filhos

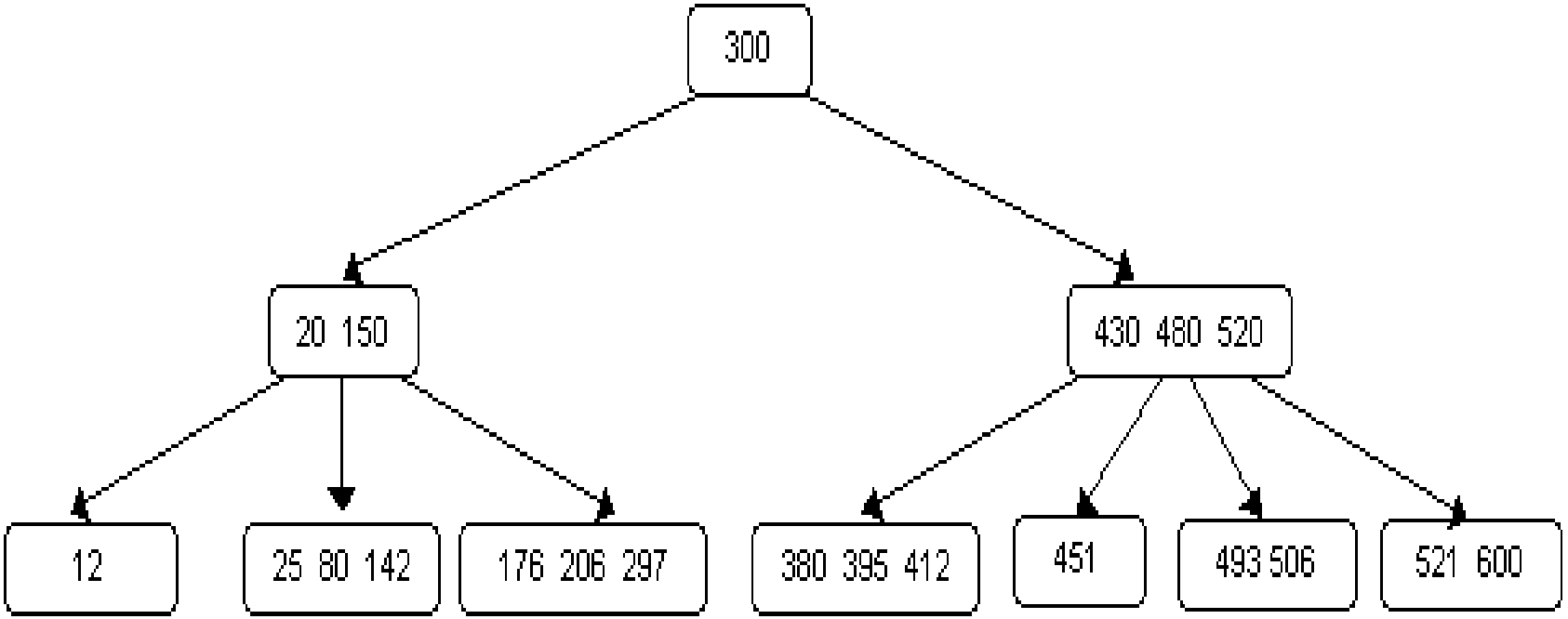
Árvores B

- Seja d um número natural. Uma árvore B de ordem d é uma árvore ordenada que é vazia, ou que satisfaz as seguintes condições:

1) A raiz é uma folha ou tem no mínimo 2 filhos;
2) Cada nó diferente da raiz e das folhas possui no mínimo (d + 1) filhos;
3) Cada nó tem no máximo (2d + 1) filhos;
4) Todas as folhas estão no mesmo nível.

Árvores B

- Um nó em uma árvore B é chamado de página
- Cada página armazena m chaves



Árvores B

Uma página não folha tem (m + 1) filhos:

- raiz - $1 \leq m \leq 2d$
- outras - $d \leq m \leq 2d$

Em cada página P as chaves estão ordenadas $s_1 \leq s_2 \leq ... \leq s_m$ e contém

- (m + 1) ponteiros $p_0, p_1, ..., p_m$ para os filhos de P
- (nas folhas estes ponteiros são NULL).

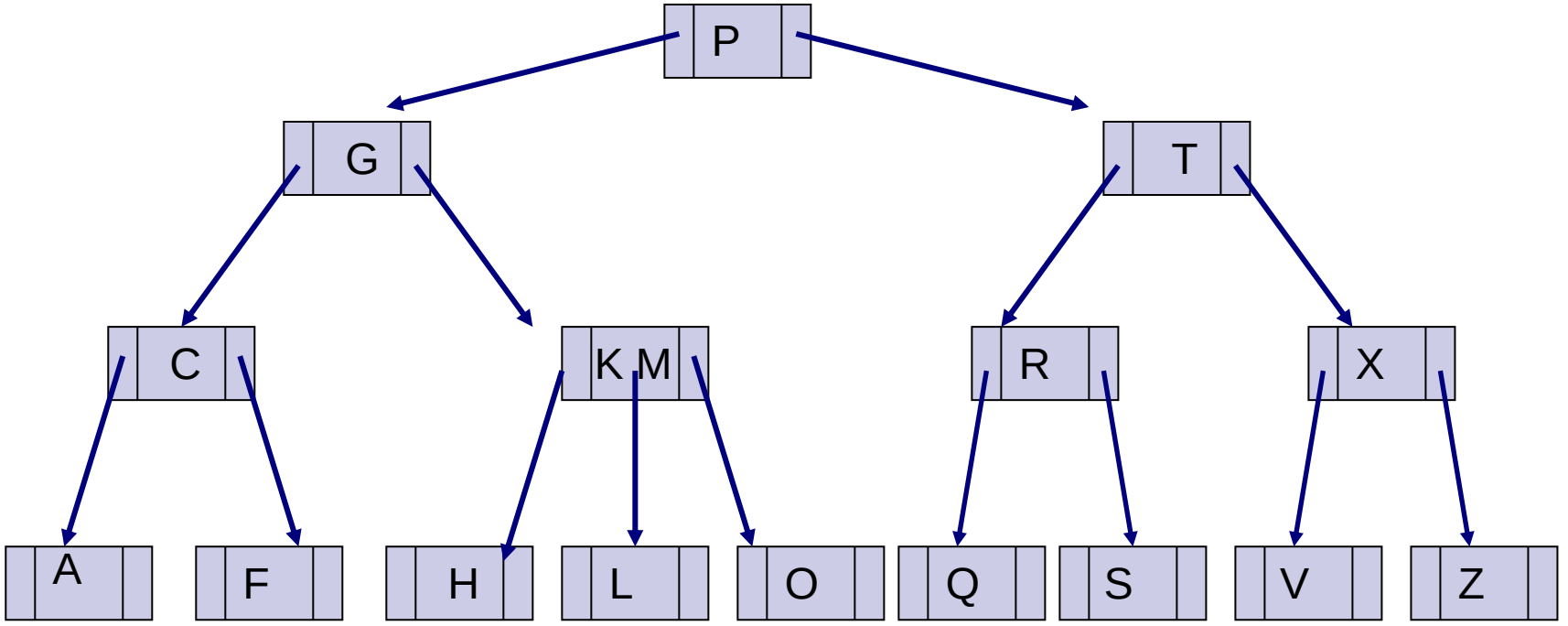
para qualquer chave y da página apontada por p_0, , $y < s_1$

Para qualquer chave y da página apontada por p_k, $1 \leq k \leq (m - 1)$, $s_k < y < s_{(k + 1)}$

Para qualquer chave y da página apontada por p_m, $y > s_m$

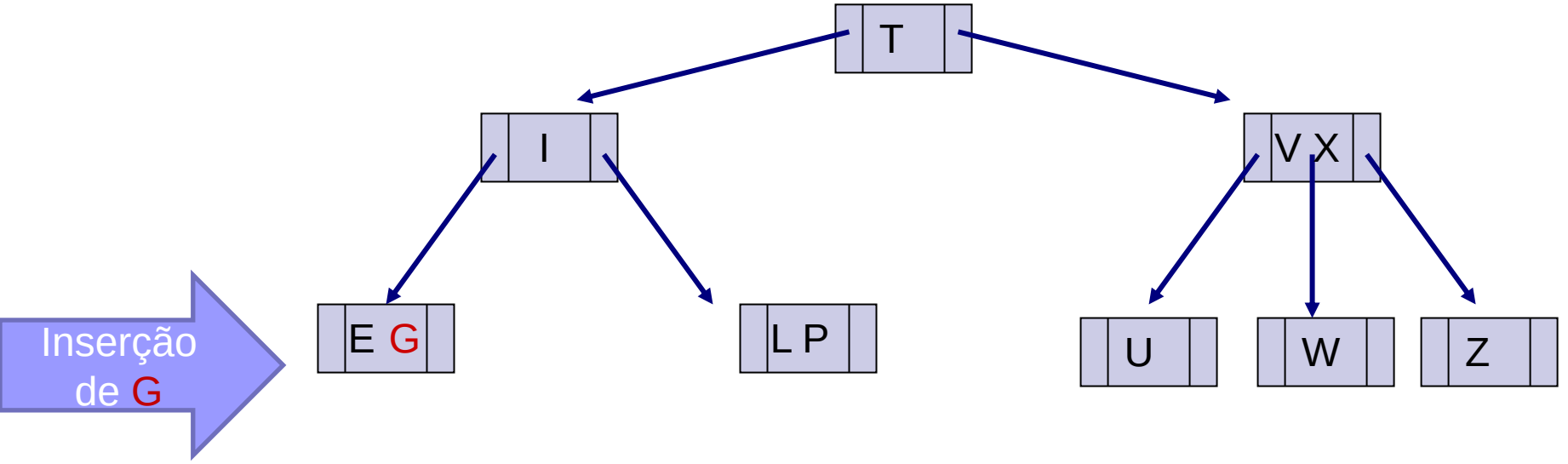
Árvores B

- Para o projeto, estamos interessados em árvores B de ordem 1 (d=1)
- Árvores 2-3



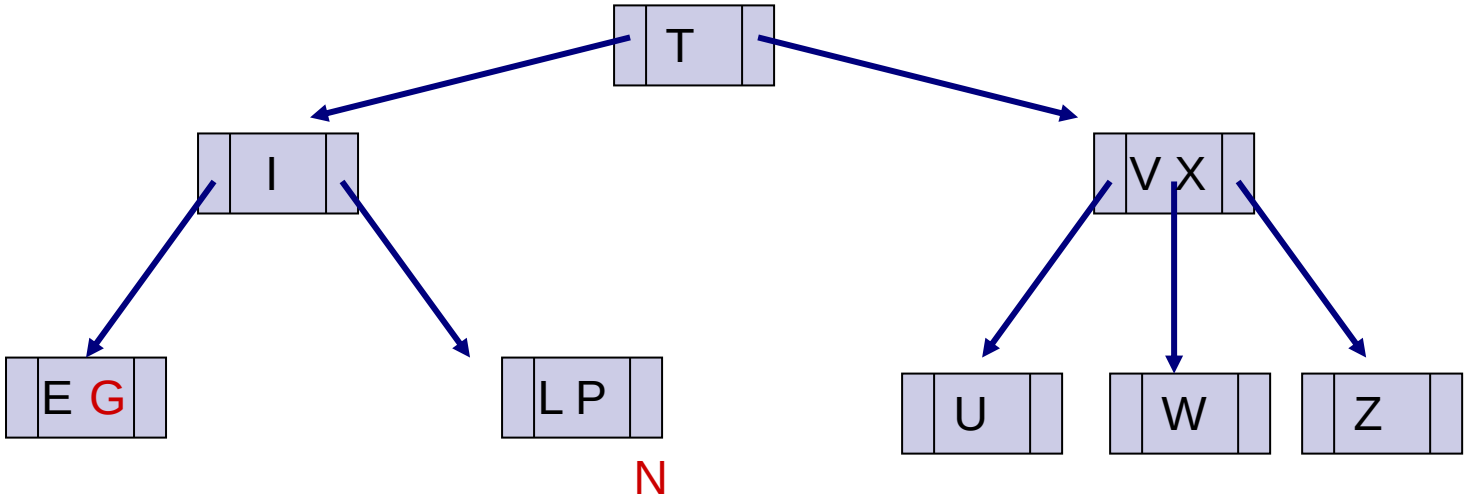
Árvores 2-3

- Inserção
  - Folha
  - Se houver espaço para acrescentar o novo elemento



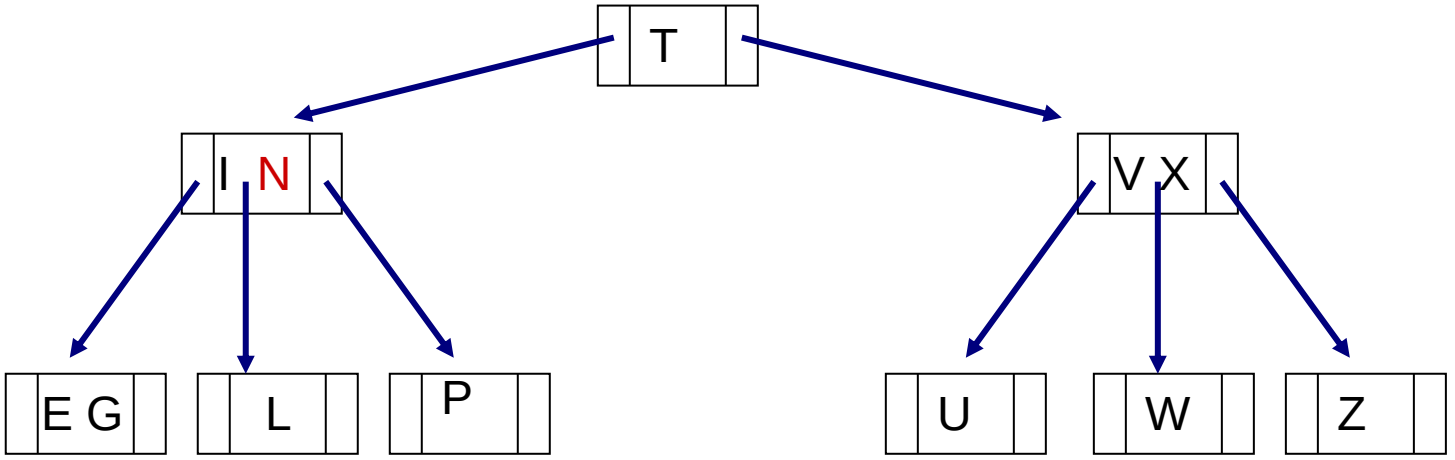
Árvores 2-3

- Se não há espaço:
  - (inserção de N)



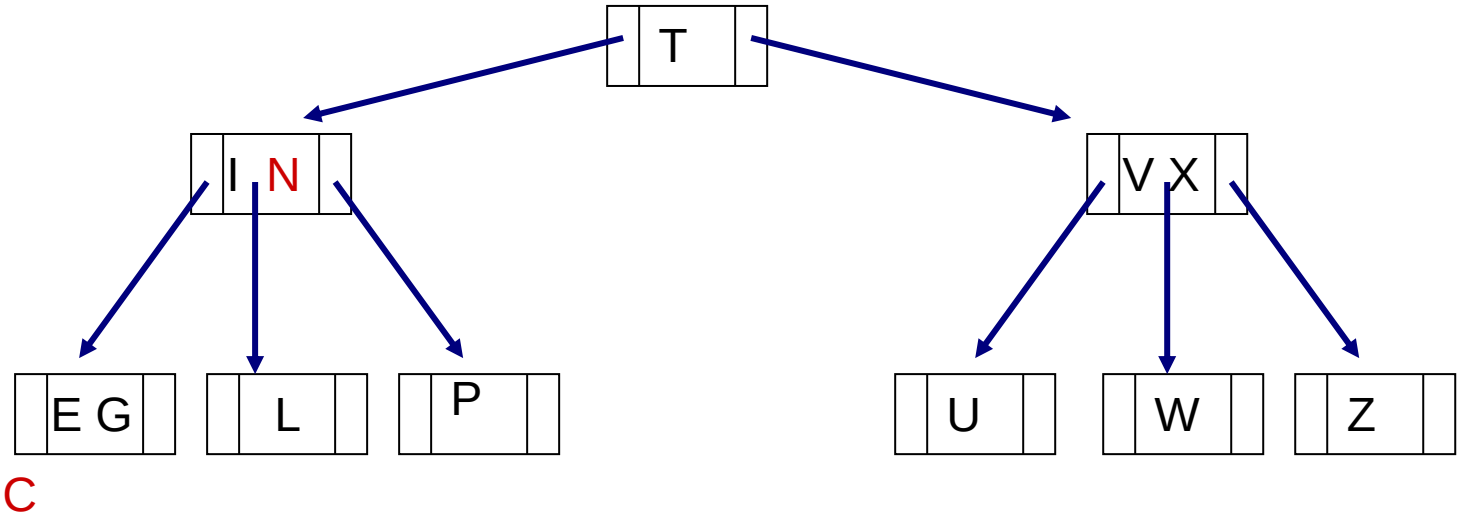
Árvores 2-3

- Gera-se dois nós com os elementos extremos do nó
- Promove-se o nó central (incluído no nível superior)



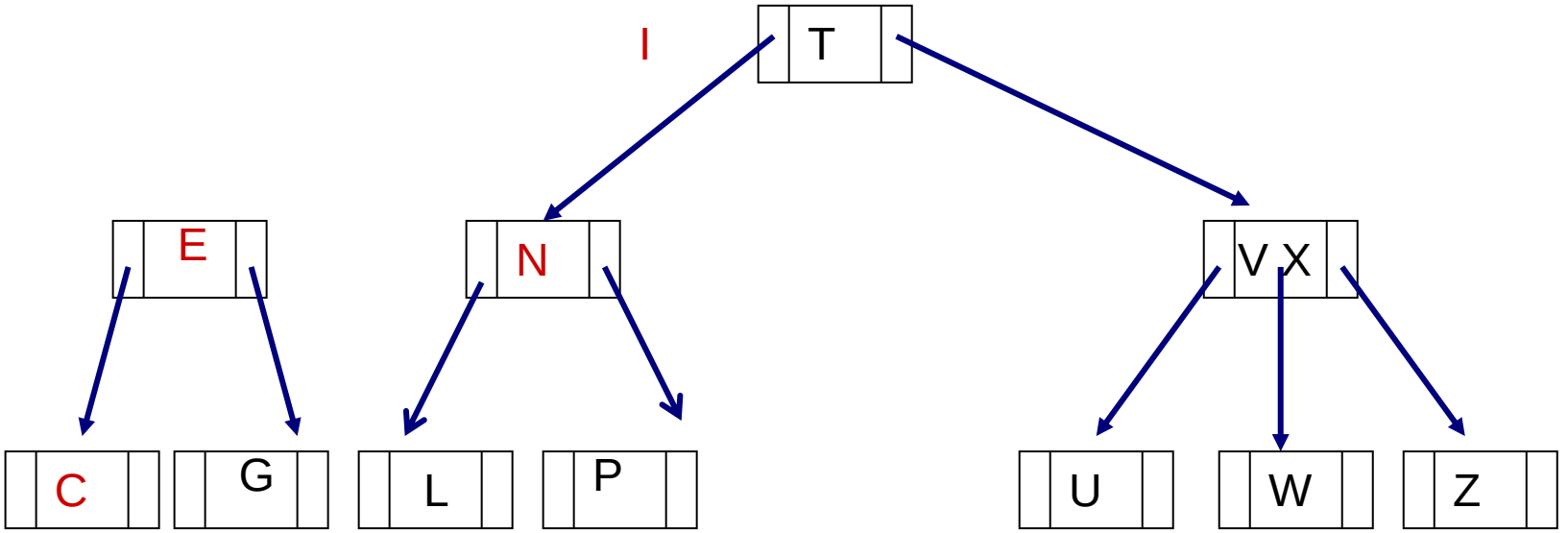
Árvores 2-3

- Se não há espaço para acrescentar o nó promovido:

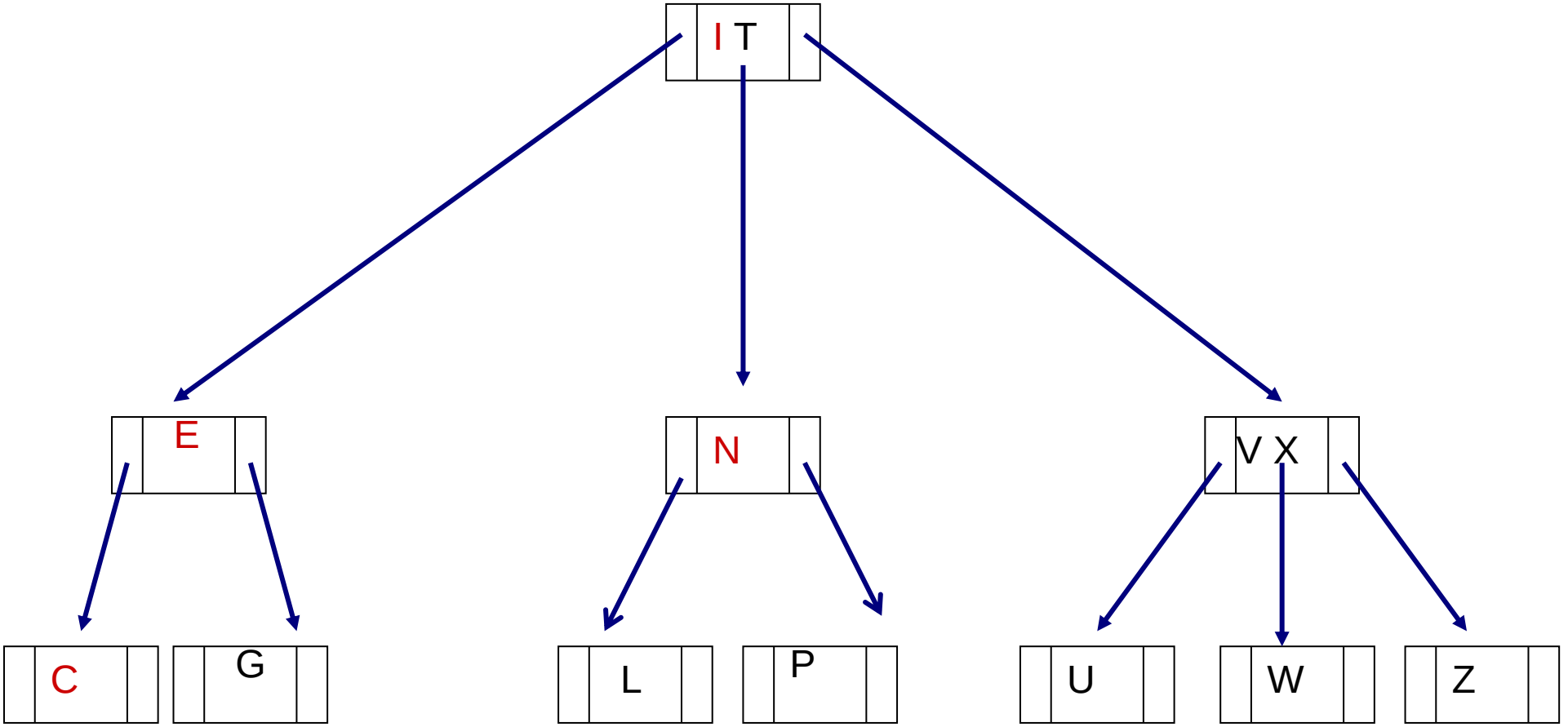


Árvores 2-3

- Mesmo procedimento
  - Nó que tenha espaço
  - Raiz



Árvores 2-3



Referências

- Cormen
- Estruturas de dados e seus algoritmos (Jayme Luiz Szwarcfiter)
- http://www.lcad.icmc.usp.br/~nonato/ED/B_arvore/btree.htm
- Katia Guim (Cin – UFPE)