



UFSM

Universidade Federal de Santa Maria



Engenharia de
CONTROLE e
AUTOMAÇÃO

Árvores

Aula 04

DPEE 1038 – Estrutura de Dados para Automação
Curso de Engenharia de Controle e Automação
Universidade Federal de Santa Maria

Prof. Rafael Concatto Beltrame
beltrame@mail.ufsm.br

Sumário

- **Aplicações de árvores binárias**
 - Procura de dados repetidos
 - Percurso de árvores binárias
 - Pré-ordem
 - Ordem
 - Pós-ordem
 - Ordenação de dados (árvore de busca binária)
 - Representação de expressões algébricas



Procura de dados repetidos

- **Utilização de árvores binárias**
 - Úteis para decisões bidirecionais em cada ponto do processo
- **Procura por dados repetidos**
 - **Solução 1:** comparar todos os dados com os que os precedem
 - **Problema:** grande número de comparações
 - **Solução 2:** uso de árvores binárias
 - O primeiro dado é colocado na raiz
 - Cada dado é comparado com o **dado da raiz**
 - **Se coincidirem** → temos uma repetição
 - **Se for menor** → examinar a subárvore esquerda
 - **Se for maior** → examinar a subárvore direita
 - **Subárvore vazia** → dado não é repetido e é gravado no nó
 - **Subárvore não-vazia** → comparar dado com a raiz da subárvore

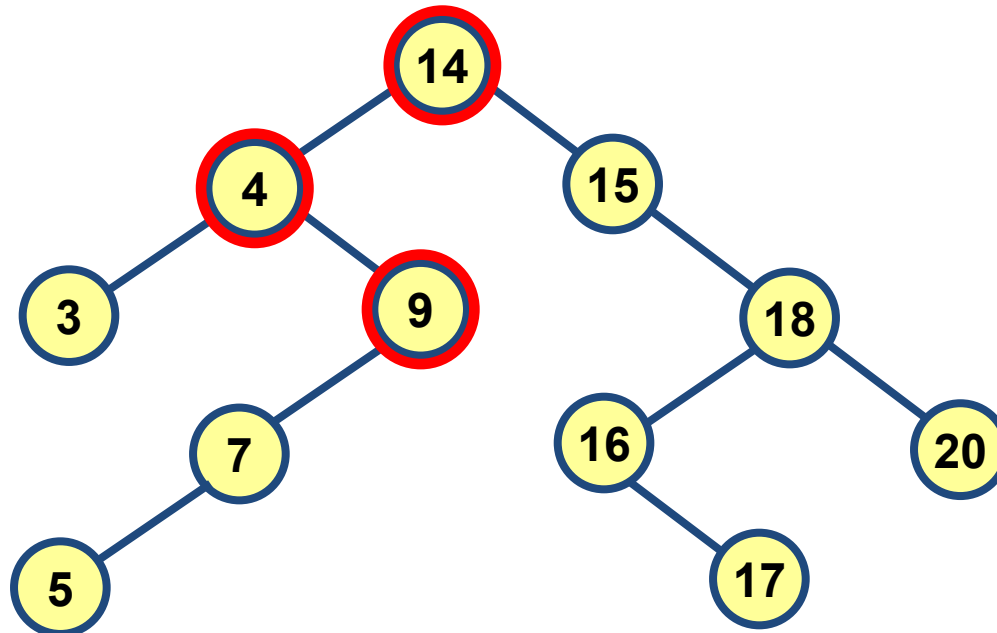
Procura de dados repetidos

- Exemplo

- Conjunto de dados:

14, 15, 4, 9, 7, 18, 3, 5, 16, 4, 20, 17, 9, 14

↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑



Procura de dados repetidos

```
scanf("%d", &x);           // Lê primeiro dado
tree = maketree(x);       // Cria raiz da árvore

while( há números na entrada ) {
    scanf("%d", &x);       // Lê novo dado
    p = tree;             // Nós auxiliares
    q = tree;

    // Obtém endereço de uma subárvore vazia direita OU
    // esquerda para salvar dado não repetido
    while (x != info(p) && q != NULL) {
        p = q;           // Percorre-se a árvore
        if (x < info(p)) { q = left(p); }
        if (x > info(p)) { q = right(p); }
    }

    // Salva dado no filho direito OU esquerdo de "p"
    if (x == info(p))
        printf("O dado %d está repetido! \n", d);
    else if (x < info(p)) { setleft(p, x); }
    else { setright(p, x); }
}
```

Percurso de árvores binárias

- **Consistem em percorrer a árvore enumerando seus nós**
 - Diferente de listas, **não existe uma ordem “natural” para os nós de uma árvore**
- **Os métodos consistem em**
 - Visitar a raiz
 - Percorrer sua subárvore esquerda
 - Percorrer sua subárvore esquerda
 - Procedimento recursivo
 - Diferem na ordem em que as operações são realizadas

Percurso de árvores binárias

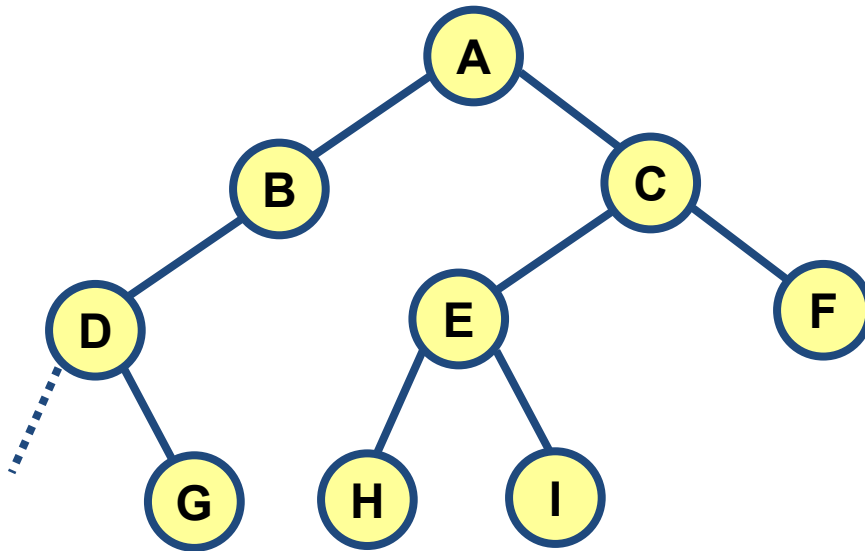
- **Percurso em pré-ordem**
 - **Percurso em profundidade**
 - 1) **Visitar a raiz**
 - 2) **Percorrer a subárvore esquerda em ordem prévia**
 - 3) **Percorrer a subárvore direita em ordem prévia**
- **Percurso em ordem**
 - **Percurso em ordem simétrica**
 - 1) **Percorrer a subárvore esquerda em ordem simétrica**
 - 2) **Visitar a raiz**
 - 3) **Percorrer a subárvore direita em ordem simétrica**

Percurso de árvores binárias

- **Percurso em pós-ordem**
 - 1) Percorrer a subárvore esquerda em ordem posterior**
 - 2) Percorrer a subárvore direita em ordem posterior**
 - 3) Visitar a raiz**

Percurso de árvores binárias

- Exemplos



Pré-ordem: **ABDGCEHIF**

Ordem: **DGBAHEICF**

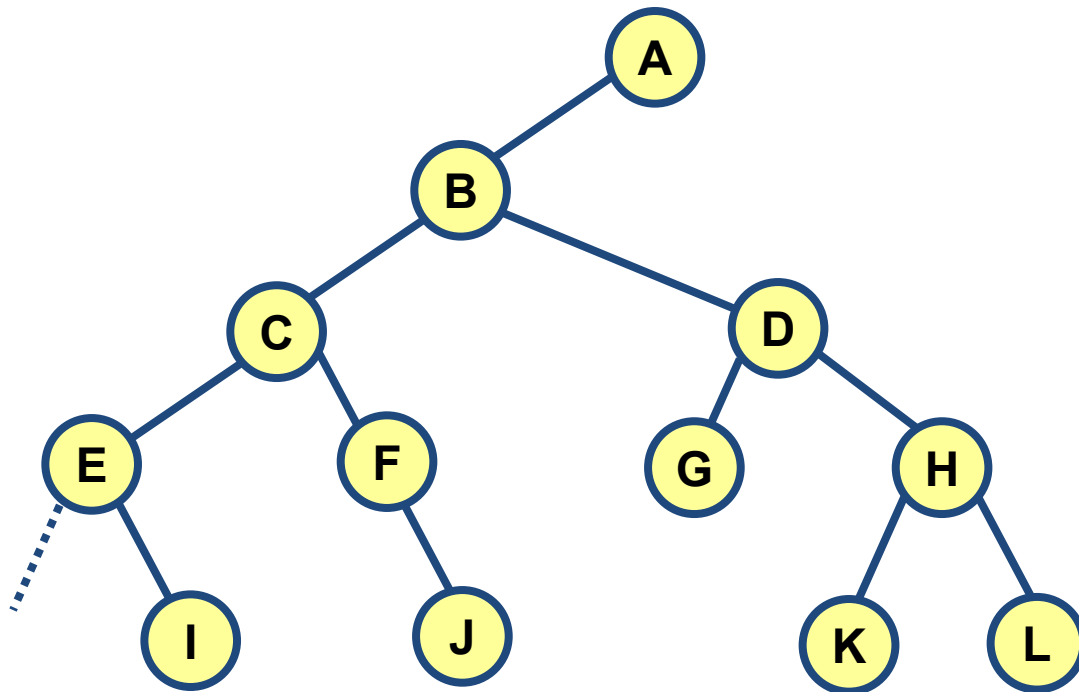
Pós-ordem: **GDBHIEFCA**

Observação

No percurso em **ordem**, “D” é o limite da subárvore esquerda, por isso é o ponto de partida. Na ausência de um filho esquerdo, a sequência de leitura manda ler a raiz.

Percurso de árvores binárias

- Exemplos



Pré-ordem: **ABCEIFJDGHKL**

Ordem: **EICFJBGDKHLA**

Pós-ordem: **IEJFCGKLHDBA**

Observação

No percurso em **ordem**, “E” é o limite da subárvore esquerda, por isso é o ponto de partida. Na ausência de um filho esquerdo, a sequência de leitura manda ler a raiz.

Árvore de busca binária

- Usualmente, algoritmos que empregam árvores binárias apresentam **duas fases distintas**
 - Construção da árvore binária
 - Percurso da árvore binária
- Seja um conjunto de dados
 - Como **classificar** os dados em **ordem crescente/decrescente**?
- **Solução**
 - O primeiro dado é colocado na raiz
 - Cada dado é comparado com o **dado da raiz**
 - **Se for menor** → examinar a subárvore esquerda
 - **Se for maior ou igual** → examinar a subárvore direita
 - Percorrer a árvore em **ordem simétrica**

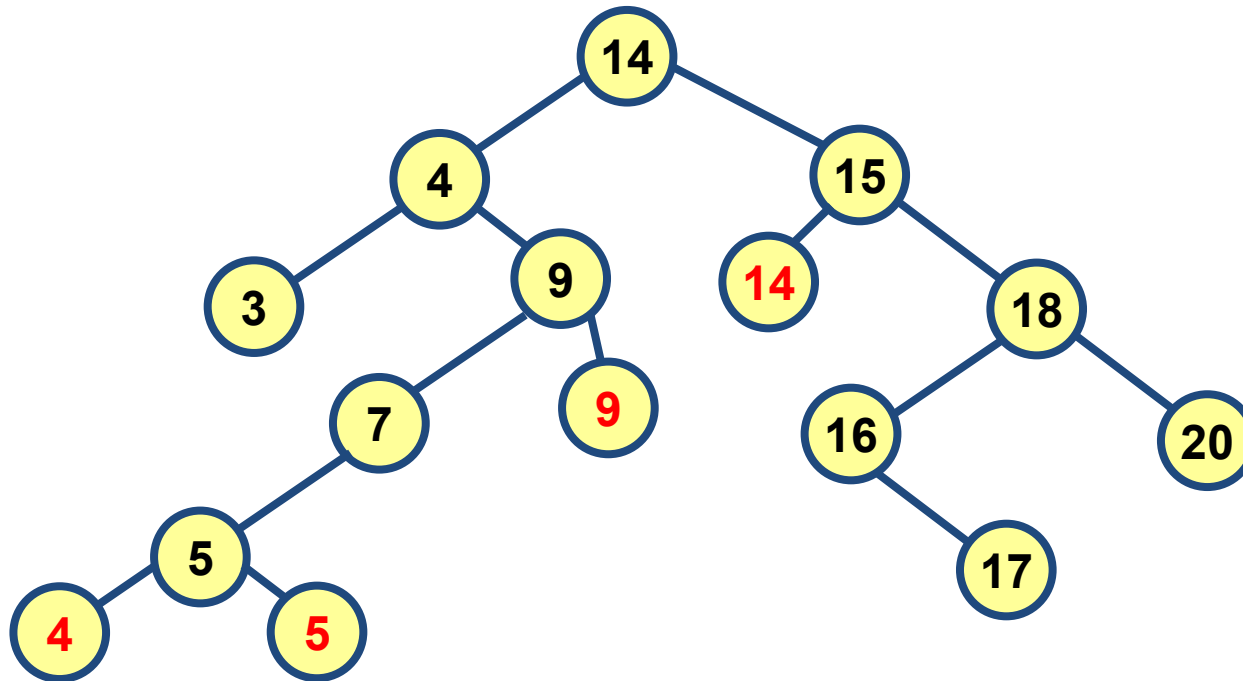
Procura de dados repetidos

- Exemplo

- Conjunto de dados:

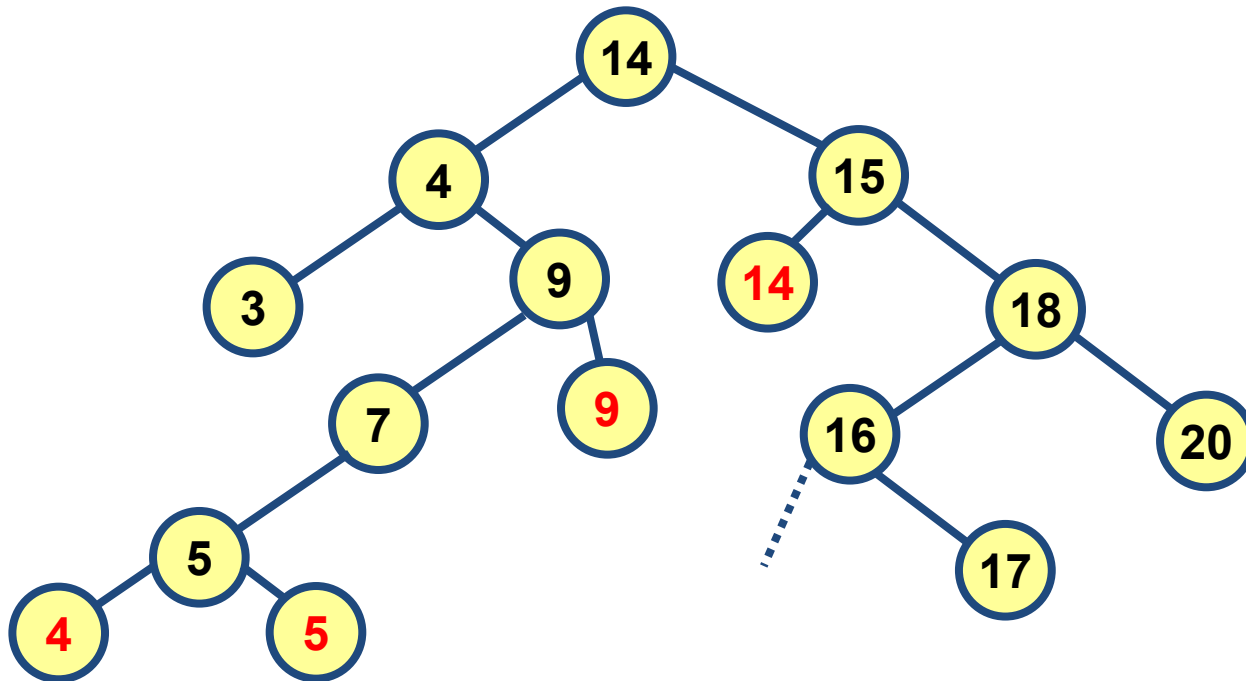
14, 15, 4, 9, 7, 18, 3, 5, 16, 4, 20, 17, 9, 14, 5

↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑



Árvore de busca binária

- Percurso em ordem (ordem simétrica)

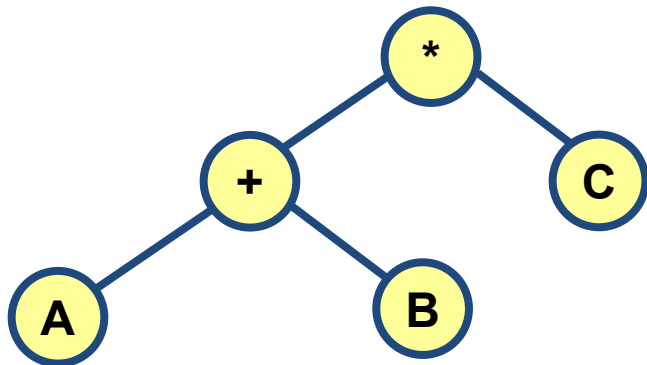


Dados: 3, 4, 4, 5, 5, 7, 9, 9, 14, 14, 15, 16, 17, 18, 20

Representação de expressões algébricas

- Seja a expressão

$$(A + B) * C$$



Leitura da árvore:

Pós-ordem: $A B + C *$



Notação pós-fixa
(RPN)