

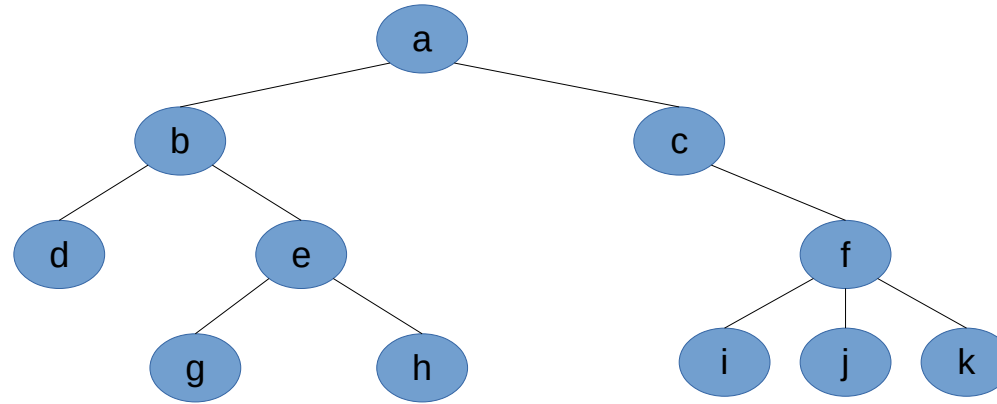
Árvore – Introdução

Prof. Flavio B. Gonzaga
flavio.gonzaga@unifal-mg.edu.br
Universidade Federal de Alfenas
UNIFAL-MG

Árvore

- Árvore é uma estrutura de dados que caracteriza a ordem e hierarquia entre seus dados.

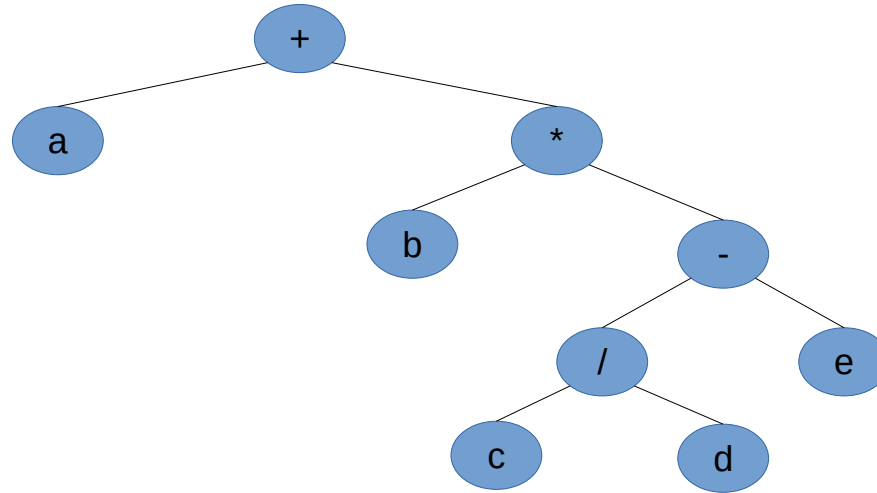
Árvore



Definições:

- Raiz: nó **a**, mas todo nó é raiz de uma subárvore.
- Grau: número de subárvores do nó, logo, **a** tem grau 2.
- Nó de grau 0: nó folha ou terminal. Ex: nó **d, g, h, i, j, k**.
- Nível de um nó: A raiz de uma árvore tem nível 1. Se um nó tem nível **i**, seus filhos terão nível **i+1**. Ex: nó **e** tem nível 3.
- Altura: é o máximo nível de seus nós. A árvore acima tem altura 4.
- Pai e filho: o nó **f**, por estar abaixo de **c**, e diretamente ligado a ele, é chamado de filho de **c**. De forma semelhante, **c** é dito pai de **f**.
- Irmão: nós com o mesmo pai são ditos irmãos. Ex: **g** e **h**.
- Nó interno: nó que não é folha.
- Floresta: conjunto de 2 ou mais árvores.

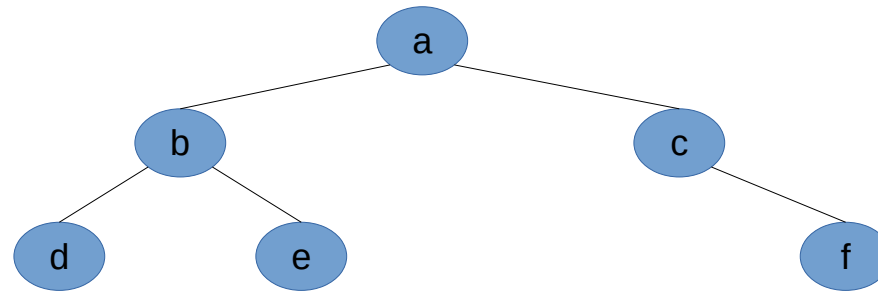
Árvore



- Nesse exemplo, utilizamos uma árvore para representar a maneira que uma expressão aritmética seria computada.
- Cada operador binário e seu par de operandos correspondem a um nó da árvore e suas duas subárvores.

Árvore Binária

- Estruturas do tipo árvore, onde o grau de cada nó é menor ou igual a dois.

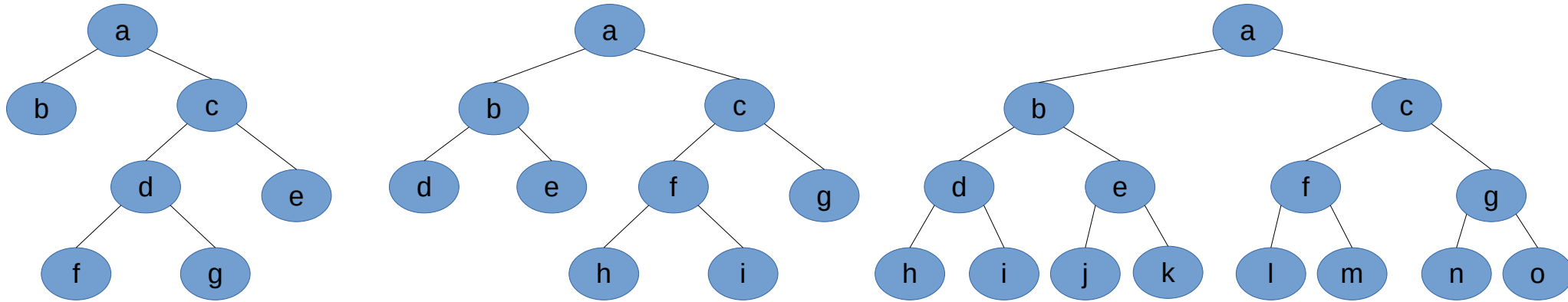


- Existe a distinção entre subárvore direita e esquerda. Para esse exemplo, a subárvore direita de **a** começa em **c**, e a subárvore esquerda de **a** começa em **b**.

Árvore Binária

- Alguns tipos especiais de árvores binárias são:
 - Estritamente binária: cada nó possui 0 ou 2 filhos.
 - Completa: se v é um nó tal que alguma subárvore de v é vazia, então v se localiza ou no último ou no penúltimo nível da árvore.
 - Cheia: se v é um nó com alguma das sua subárvores vazias, então v se localiza no último nível.

Árvore Binária

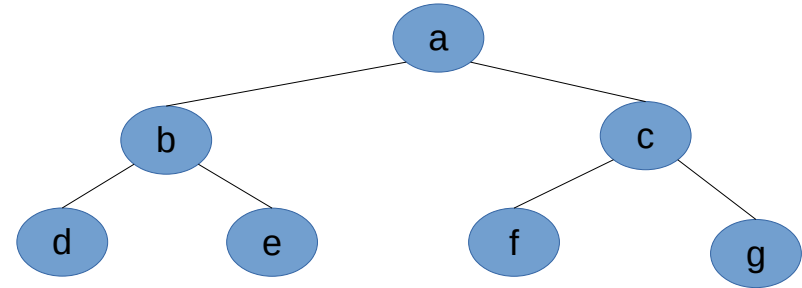
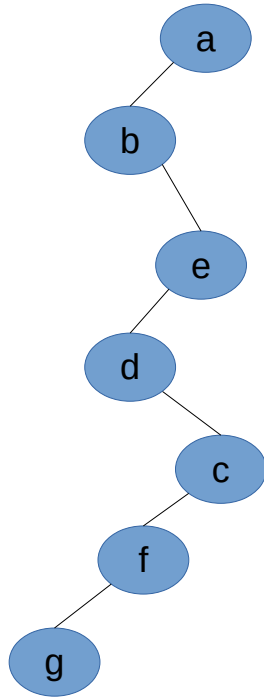


- Estritamente binária: cada nó possui 0 ou 2 filhos.
- Completa: se **v** é um nó tal que alguma subárvore de **v** é vazia, então **v** se localiza ou no último ou no penúltimo nível da árvore.
- Cheia: se **v** é um nó com alguma das sua subárvores vazias, então **v** se localiza no último nível.

Árvore Binária

- A relação entre a altura de uma árvore binária e seu número de nós é um dado importante para várias aplicações.
 - Para um valor fixo de quantidade de nós (n), quais são as árvores que possuem altura (h) máxima e mínima?
 - Altura máxima será aquela cujos nós anteriores possuem exatamente uma subárvore vazia (denominadas zigue-zague).
 - Uma árvore completa apresenta a altura mínima.

Árvore Binária

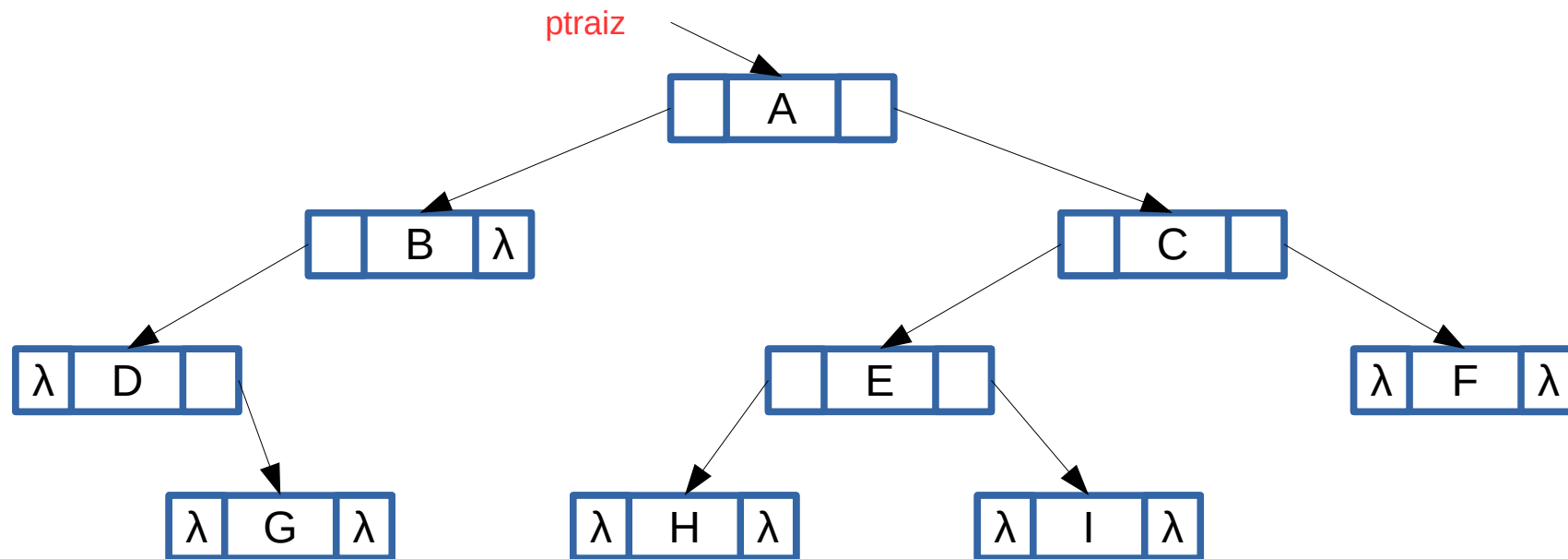


- Para um valor fixo de quantidade de nós (**n**), quais são as árvores que possuem altura (**h**) máxima e mínima?

Árvore Binária

- O armazenamento de uma árvore binária surge naturalmente de sua definição.
- Cada nó deve possuir dois campos de ponteiros, **esq** e **dir**, que apontam para suas subárvores esquerda e direita respectivamente.
- O ponteiro **ptr** indica a raiz da árvore.
- Os campos do nó da árvore que contêm as informações relacionadas ao problema serão representados como um só campo de nome **info**.

Árvore Binária



λ = NULO

Árvore Binária - Caminhamento

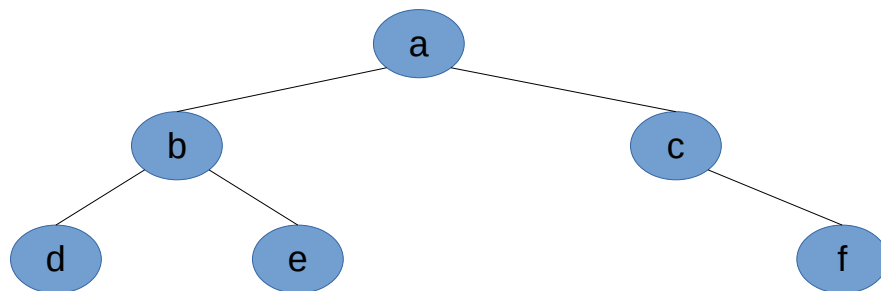
- Caminhar em uma árvore binária significa percorrer todos os nós de forma sistemática, de modo que cada nó seja visitado uma vez.
 - Visitar um nó significa operar, de alguma forma, com a informação à ele relativa.
 - No processo de percorrer uma árvore pode ser necessário passar várias vezes por alguns de seus nós, sem visitá-los.

Árvore Binária - Caminhamento

- Usando-se de recursividade, existem 3 formas básicas de caminhamento em árvore binária:
 - Pré-ordem:
 - 1. processa o nó.
 - 2. percorre recursivamente, em pré-ordem, a subárvore esquerda.
 - 3. percorre recursivamente, em pré-ordem, a subárvore direita.
 - Em-ordem:
 - 1. percorre recursivamente, em ordem, a subárvore esquerda.
 - 2. processa o nó.
 - 3. percorre recursivamente, em ordem, a subárvore direita.
 - Pós-ordem:
 - 1. percorre recursivamente, em pós-ordem, a subárvore esquerda.
 - 2. percorre recursivamente, em pós-ordem, a subárvore direita.
 - 3. processa o nó.

Árvore Binária - Caminhamento

- Pré-ordem:
 - 1. processa o nó;
 - 2. percorre recursivamente, em pré-ordem, a subárvore esquerda.
 - 3. percorre recursivamente, em pré-ordem, a subárvore direita.
- Em-ordem:
 - 1. percorre recursivamente, em ordem, a subárvore esquerda.
 - 2. processa o nó.
 - 3. percorre recursivamente, em ordem, a subárvore direita.
- Pós-ordem:
 - 1. percorre recursivamente, em pós-ordem, a subárvore esquerda.
 - 2. percorre recursivamente, em pós-ordem, a subárvore direita.
 - 3. processa o nó.



Pré-ordem: a, b, d, e, c, f.
Em-ordem: d, b, e, a, c, f.
Pós-ordem: d, e, b, f, c, a.

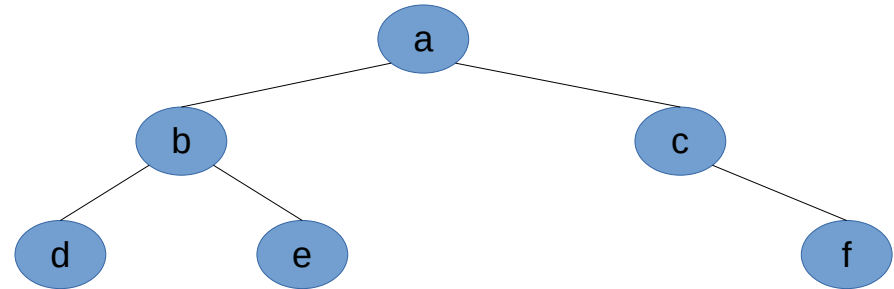
Árvore Binária – Caminhamento

```
pre-ordem(pt){  
  visita(pt);  
  se pt^.esq  $\neq$   $\lambda$  então pre-ordem(pt^.esq)  
  se pt^.dir  $\neq$   $\lambda$  então pre-ordem(pt^.dir)  
}
```

```
em-ordem(pt){  
  se pt^.esq  $\neq$   $\lambda$  então em-ordem(pt^.esq)  
  visita(pt);  
  se pt^.dir  $\neq$   $\lambda$  então em-ordem(pt^.dir)  
}
```

```
pos-ordem(pt){  
  se pt^.esq  $\neq$   $\lambda$  então pos-ordem(pt^.esq)  
  se pt^.dir  $\neq$   $\lambda$  então pos-ordem(pt^.dir)  
  visita(pt);  
}
```

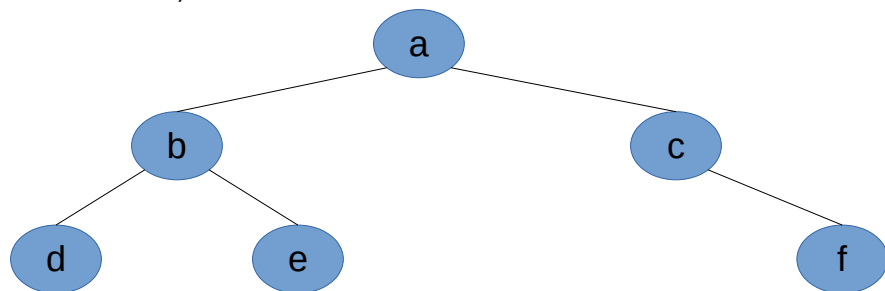
```
visita(pt){  
  imprime(pt^.info);  
}
```



Árvore Binária – Altura de um nó

```
pre-ordem(pt){  
  visita(pt);  
  se pt^.esq ≠ λ então pre-ordem(pt^.esq)  
  se pt^.dir ≠ λ então pre-ordem(pt^.dir)  
}  
  
em-ordem(pt){  
  se pt^.esq ≠ λ então em-ordem(pt^.esq)  
  visita(pt);  
  se pt^.dir ≠ λ então em-ordem(pt^.dir)  
}  
  
pos-ordem(pt){  
  se pt^.esq ≠ λ então pos-ordem(pt^.esq)  
  se pt^.dir ≠ λ então pos-ordem(pt^.dir)  
  visita(pt);  
}
```

```
visita(pt){  
  se pt^.esq ≠ λ então  
    alt1 := (pt^.esq)^.altura;  
  senão alt1 := 0;  
  
  se pt^.dir ≠ λ então  
    alt2 := (pt^.dir)^.altura;  
  senão alt2 := 0;  
  
  se alt1 > alt2 então  
    pt^.altura := alt1 + 1;  
  senão  
    pt^.altura := alt2 + 1;  
}
```



Qual dos métodos de caminhamento que, ao ser usado juntamente com a função visita, produziria o resultado de calcular a altura do nó?

Referências Bibliográficas

- Estruturas de Dados e Seus Algoritmos. Szwarcfiter J. L.; Markenzon L.. 3a Edição. Editora LTC. 2010.
- Estruturas De Dados Usando C. Tenenbaum A. M.; Langsam Y.; Augenstein M. J.. 1a Edição. Editora Pearson. 1995.
- Introdução a Estruturas de Dados: Com Técnicas de Programação em C. Celes W.; Cerqueira R.; Rangel J.. 2a Edição. Editora Elsevier. 2017.