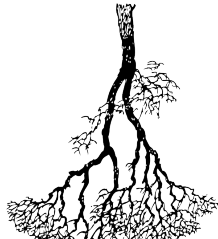


Algoritmos e Estrutura de Dados

Árvores Binárias

prof. Frederico Santos de Oliveira

Universidade Federal de Mato Grosso
Faculdade de Engenharia



Roteiro

1 Objetivos

2 Introdução

3 Árvore Binária

4 Implementação

- Estrutura Nodo
- Estrutura Árvore Binária
- Algoritmo CriaÁrvore — ÁrvoreVazia
- Algoritmo QuantidadeNodos
- Algoritmo AlturaÁrvore
- Percurso em Árvore

Objetivos

Esta aula tem como objetivos:

- 1 Apresentar os conceitos básicos sobre árvores;
- 2 Exemplificar os algoritmos de manipulação de árvores por meio de pseudo-códigos.

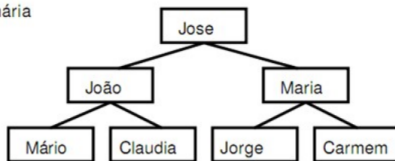
Introdução

- Uma árvore é uma abstração matemática usada para representar estruturas hierárquicas não lineares dos objetos modelados.
- É definida usando um conjunto de nodos (ou vértices) e arestas, que são utilizadas para conectar qualquer par de nodos ou vértices.
- Basicamente, qualquer problema em que exista algum tipo de hierarquia pode ser representado por uma árvore.

Introdução

Exemplo

Arvore Binária

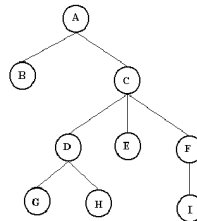
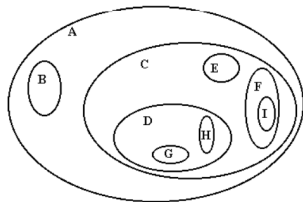


Arvore Ternaria



Formas de Visualização

Representação Hierárquica



Grafo

Uma árvore é um tipo especial de grafo: um grafo **não-direcionado**, **conexo** e **acíclico**.

- Uma árvore é formada por um conjunto de nodos ligados por arestas de forma hierárquica, simulando as árvores encontradas na natureza.
- Porém, nessa estrutura as árvores crescem de cima para baixo, ou seja, a raiz se encontra no topo e as folhas na parte mais baixa.

Introdução

Principais Conceitos

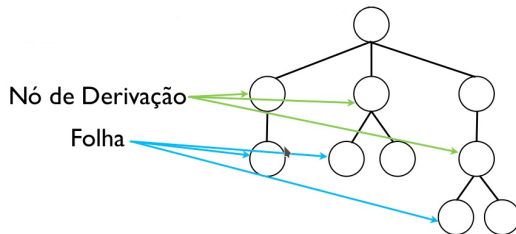
A seguir, a nomenclatura usada para se trabalhar com uma árvore:

- **Raiz:** é o nodo localizado na parte mais alta da árvore, o único que não possui pai.
- **Pai:** também chamado de ancestral, é o nodo antecessor imediato de outro nodo.
- **Filho:** é o nodo sucessor imediato de outro nodo.

Introdução

Principais Conceitos

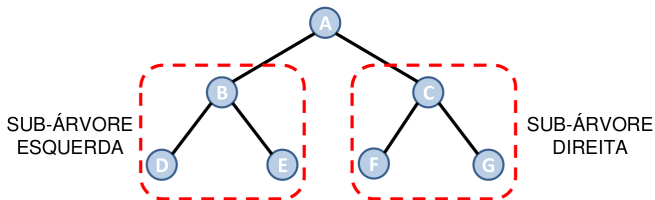
- **Nodo folha:** também chamado de nodo terminal, é qualquer nodo que não possui filhos.
- **Nodo de derivação:** também chamado de nodo não-terminal, ou nodo interno, é qualquer nodo que possui ao menos um filho.



Principais Conceitos

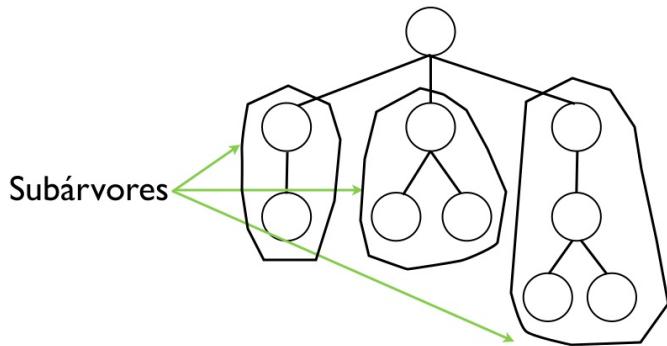
- **Subárvores:**

- Dado um nodo da árvore, cada filho seu é considerado a raiz de uma nova subárvores.
- Qualquer nodo é a raiz de uma subárvore.



Introdução

Principais Conceitos



Introdução

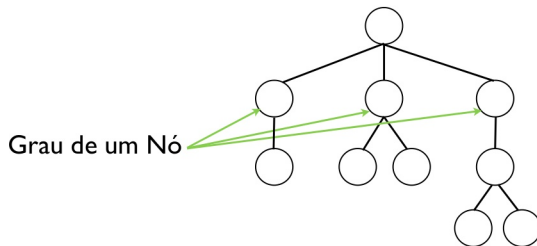
Principais Conceitos

- **Grau de um nodo:**

- O grau de um nodo é dado pelo número de subárvores que ele possui.

- **Grau de uma árvore:**

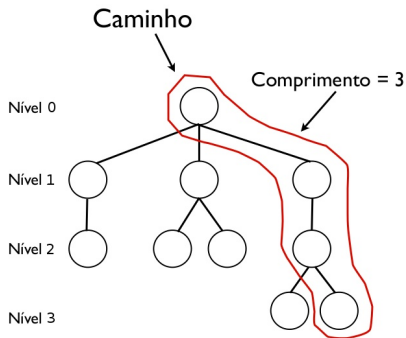
- É o maior valor dentre os graus de todos os seus nodos.



Introdução

Principais Conceitos

- **Caminho:** é uma sequência de nodos de modo que existe sempre uma aresta ligando o nodo anterior com o seguinte.
- **Nível:** em uma árvore os nodos são classificados em diferentes níveis, começando a partir do nodo raiz (nível 0).
- **Comprimento de um caminho:** quantidade de arestas em um caminho.

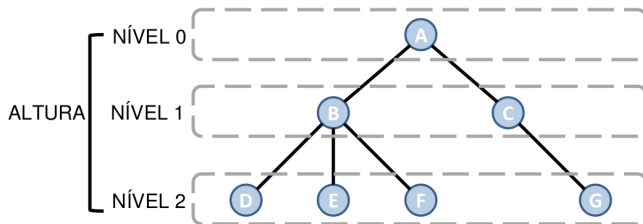


- **Altura:**

- A altura (ou profundidade) de uma árvore é igual ao nível máximo de seus nodos.
- Ou seja, é o comprimento do caminho mais longo da raiz até uma das suas folhas.
- A altura de um nodo é igual ao comprimento do caminho do nodo até a raiz.

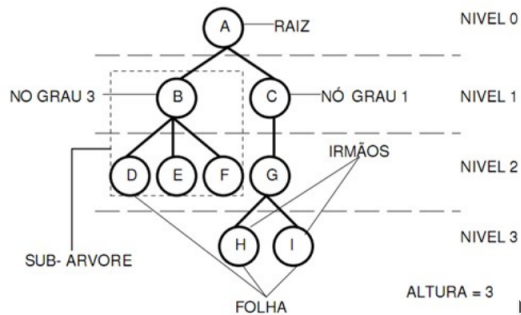
Introdução

Principais Conceitos



Introdução

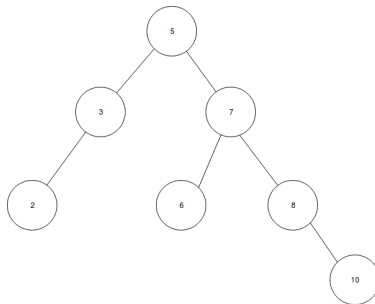
Principais Conceitos



Árvore Binária

Exercício

Considere a árvore a seguir:

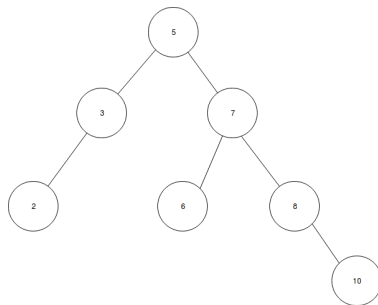


- Qual a altura do nodo com valor 6?
- Qual o grau do nodo com valor 3?
- Qual a altura da árvore?
- Qual o nível do nodo com valor 3?
- Qual o comprimento do caminho que liga o nodo 5 ao 8?

Árvore Binária

Exercício

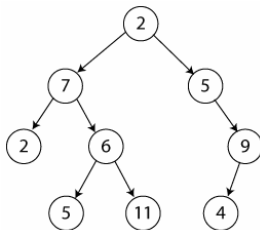
Considere a árvore a seguir:



- Qual a altura do nodo com valor 6? **Altura = 2**
- Qual o grau do nodo com valor 3? **Grau = 1**
- Qual a altura da árvore? **Altura = 3**
- Qual o nível do nodo com valor 3? **Nível = 1**
- Qual o comprimento do caminho que liga o nodo 5 ao 8? **Comp. = 2**

Árvore Binária

- Uma árvore binária é um tipo especial de árvore em que cada nodo pode possuir no máximo duas subárvores: a subárvore da **esquerda** e a da **direita**.
- Caso o nodo possua nenhuma subárvore, este será um nodo folha.
- Exemplo:



Árvore Binária

Classificação

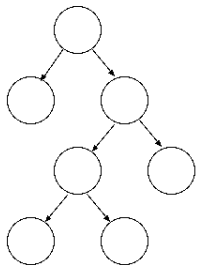
Tipos de Árvore Binária

- **Estritamente Binária:** é uma árvore binária em que cada nodo tem 0 ou 2 filhos.
- **Completa:** é aquela em se n é um nodo com algumas de subárvores vazias, então n se localiza no penúltimo ou no último nível.
- **Cheia:** é uma árvore em que se um nodo tem alguma subárvore vazia então ele está no último nível.

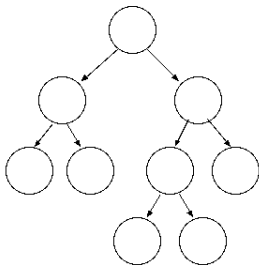
Árvore Binária

Classificação

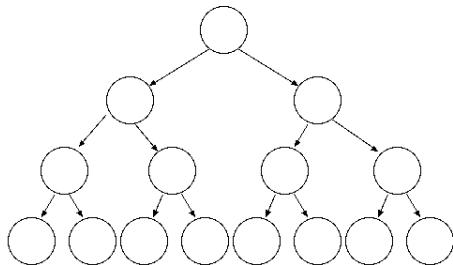
Exemplos:



Árvore estritamente binária



Árvore binária completa

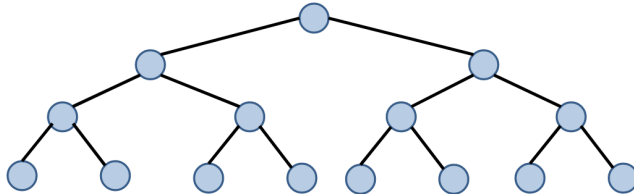


Árvore binária cheia

Árvore Binária

Classificação

Árvore Binária Cheia



| Nível (n) | 0 | 1 | 2 | 3 |
|-----------|-------|-------|-------|-------|
| Nº de nós | 1 | 2 | 4 | 8 |
| Potência | 2^0 | 2^1 | 2^2 | 2^3 |

Árvore Binária

Classificação

Na árvore cheia é possível calcular o número de nodos por nível, assim como o número total de nodos:

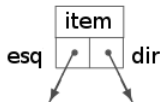
- Um nível i possui exatamente 2^i nodos.
- Se um nível i possui n nodos, o nível $i + 1$ possui $2n$ nodos.
- Uma árvore de altura h possui $2^{h+1} - 1$ nodos.

Árvore Binária

Implementação

A estrutura **nodo** contém três campos:

- Um ponteiro **esq**, que indica o filho da esquerda daquele nodo.
- Um ponteiro **dir**, que indica o filho da direita daquele nodo.
- Um campo **item** do tipo **int**, que é o tipo de dado a ser armazenado no nodo da árvore.



Árvore Binária

Estrutura Nodo

Algoritmo 1: Nodo

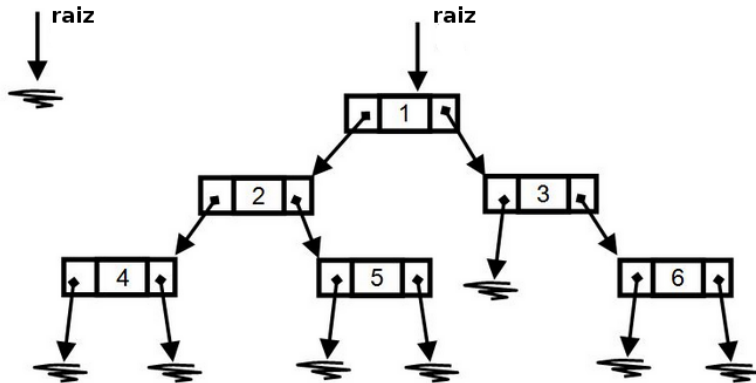
```
1 início
2   registro {
3     Inteiro: item;
4     Ponteiro Nodo: esq;
5     Ponteiro Nodo: dir;
6   } Nodo;
```

Árvore Binária

Implementação

A estrutura **árvore** trata-se de um ponteiro do tipo **nodo**:

- O ponteiro **raiz** aponta para o nodo raiz da árvore.
- Se a árvore está vazia, o ponteiro **raiz** aponta para NULL.



Árvore Binária

Implementação

Algoritmo 2: CriaÁrvore

Entrada: Ponteiro r para raiz.

```
1 início
2    $r \leftarrow \text{NULL}$ 
```

Algoritmo 3: ÁrvoreVazia

Entrada: Ponteiro r para raiz.

Saída: V ou F

```
1 início
2   se  $(r = \text{NULL})$  então
3     retorna V
4   senão
5     retorna F
```

Árvore Binária

Implementação

A quantidade de nodos em uma árvore com raiz em r é calculada da seguinte forma:

- Verifica se r é vazio.
 - Caso sim, a quantidade de nodos é zero.
 - Caso contrário, o total de nodos da árvore com raiz em r será a soma da quantidade de nodos em suas subárvores mais um (referente ao nodo raiz r).
- A seguir, o pseudocódigo.

Árvore Binária

Implementação

Algoritmo 4: QuantidadeNodos

Entrada: Ponteiro r para raiz.

Saída: Quantidade de nodos em r .

```
1 início
2   se ( $r = NULL$ ) então
3     retorna 0
4   senão
5      $total\_esq \leftarrow QuantidadeNodos(r.esq)$ 
6      $total\_dir \leftarrow QuantidadeNodos(r.dir)$ 
7     retorna  $total\_esq + total\_dir + 1$ 
```

Adaptado de (BACKES, 2016)

Árvore Binária

Implementação

A altura é calculada da seguinte forma:

- Verifica se o nodo é um nodo folha.
 - Caso sim, sua altura é zero. Nesse caso, a altura de seus filhos é -1.
 - Caso contrário, obtém-se a maior altura entre suas subárvores, e incrementa em um.
- A seguir, o pseudocódigo.

Árvore Binária

Implementação

Algoritmo 5: AlturaÁrvore

Entrada: Ponteiro r para raiz.

Saída: Altura da árvore com raiz em r .

```
1 início
2   se  $(r = NULL)$  então
3     retorna -1
4   senão
5     alt_esq  $\leftarrow$  AlturaÁrvore( $r.esq$ )
6     alt_dir  $\leftarrow$  AlturaÁrvore( $r.dir$ )
7     se  $(alt\_esq > alt\_dir)$  então
8       retorna alt_esq + 1
9     senão
10      retorna alt_dir + 1
```

Árvore Binária

Percurso em Árvore

- Não existe uma ordem “natural” para se percorrer todos os de uma árvore binária.
- Apesar disso, existem algumas formas muito utilizadas para se percorrer uma árvore.
- São elas:
 - Percurso **pré-ordem**: visita a raiz, o filho da esquerda e o filho da direita.
 - Percurso **em-ordem**: visita o filho da esquerda, a raiz e o filho da direita.
 - Percurso **pós-ordem**: visita o filho da esquerda, o filho da direita e a raiz.

Árvore Binária

Implementação

Algoritmo 6: PercursoPréOrdem

Entrada: Ponteiro r para raiz.

```
1 início
2   se ( $r \neq NULL$ ) então
3     Imprima (r.item)
4     PercursoPréOrdem(r.esq)
5     PercursoPréOrdem(r.dir)
```

Adaptado de (BACKES, 2016)

Árvore Binária

Implementação

Algoritmo 7: PercursoEmOrdem

Entrada: Ponteiro r para raiz.

```
1 início
2   se ( $r \neq NULL$ ) então
3     PercursoEmOrdem(r.esq)
4     Imprima (r.item)
5     PercursoEmOrdem(r.dir)
```

Adaptado de (BACKES, 2016)

Árvore Binária

Implementação

Algoritmo 8: PercursoPósOrdem

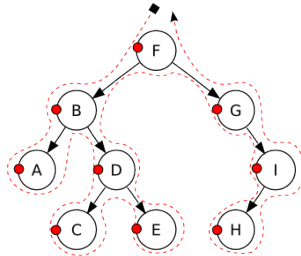
Entrada: Ponteiro r para raiz.

```
1 início
2   se ( $r \neq NULL$ ) então
3     PercursoPósOrdem(r.esq)
4     PercursoPósOrdem(r.dir)
5     Imprima (r.item)
```

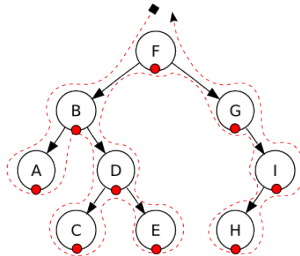
Adaptado de (BACKES, 2016)

Árvore Binária

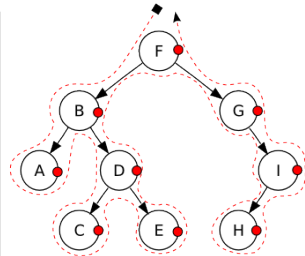
Percurso em Árvore



Pré-ordem: F, B, A, D, C, E, G, I, H



Ordem simétrica: A, B, C, D, E, F, G, H, I

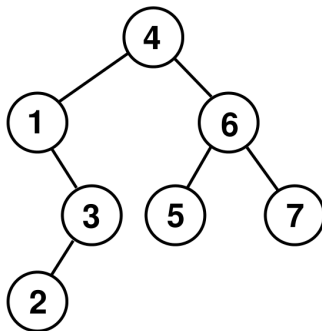


Pós-ordem: A, C, E, D, B, H, I, G, F

Árvore Binária

Percurso em Árvore - Exercício

Considere a árvore a seguir:



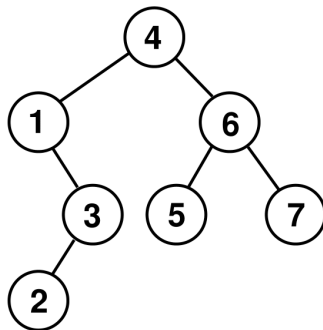
Qual a ordem de visitação dos nodos utilizando o percurso:

- Pré-ordem.
- Em-ordem.
- Pós-ordem.

Árvore Binária

Percurso em Árvore - Exercício

Considere a árvore a seguir:




Qual a ordem de visitação dos nodos utilizando o percurso:

- Pré-ordem. **4, 1, 3, 2, 6, 5, 7**
- Em-ordem. **1, 2, 3, 4, 5, 6, 7**
- Pós-ordem. **2, 3, 1, 5, 7, 6, 4**

Referências

Bibliografia Básica

- Bibliografia Básica

 BACKES, A. *Estrutura de Dados Descomplicada - Em Linguagem C*. 1. ed. São Paulo: Elsevier, 2016.

- Material Complementar

- Link código-fonte e listas de exercícios- Material disponível on-line

- Vídeo-aulas

- Curso de programação e estrutura de dados em linguagem C
- Estrutura de dados - Univesp

Algoritmos e Estrutura de Dados

Árvores Binárias

prof. Frederico Santos de Oliveira

Universidade Federal de Mato Grosso
Faculdade de Engenharia

