Árvores

Prof. Marcel Hugo
Estruturas de Dados

Departamento de Sistemas e Computação
Universidade Regional de Blumenau – FURB

Slides criados a partir do material Profa. Patricia Dockhorn Costa, disciplina de Estrutura de Dados (UFES); Prof. David Menotti, disciplina de Algoritmos e Estruturas de Dados I, DECOM – UFOP; e do Prof. Paulo Rodacki Gomes, Disciplina de Algoritmo e Estrutura de Dados, DSC - FURB

1

Introdução - Árvore

- Estruturas estudadas até agora não são adequadas para representar dados que devem ser dispostos de maneira hierárquica
- Exemplo:
  - hierarquia de pastas no sistema operacional
  - Árvore genealógica
  - Organograma de uma empresa
- Árvores são estruturas adequadas para representação de hierarquias

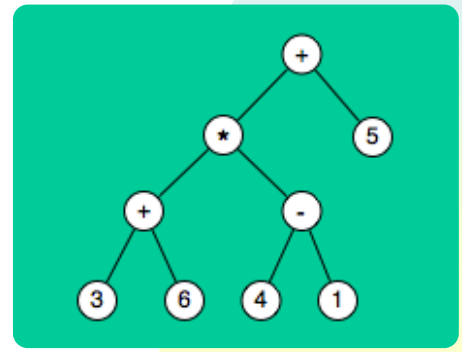
2

Introdução - Árvore

- Exemplos de dados em hierarquias
  - hierarquia de pastas

  - Expressões aritméticas

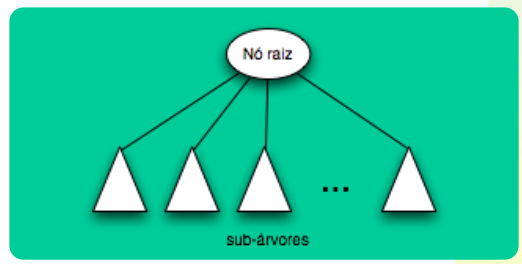
    (3+6)*(4-1)+5



3

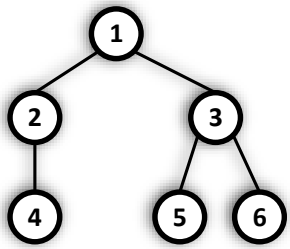
Introdução - Árvore

- Árvore é um conjunto de nós tal que
  - existe um nó r, denominado **raiz** com zero ou mais sub-árvores, cujas raízes estão ligadas a r
  - os nós raízes destas sub-árvores são os ***filhos*** de r
  - os ***nós internos*** da árvore são os nós com filhos
  - as ***folhas*** ou *nós externos* são os nós sem filhos



4

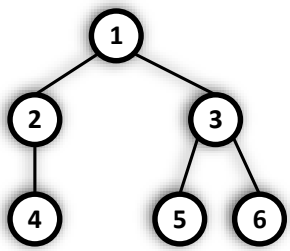
Conceitos Básicos



- **Nó raiz**
  - É o nó que está na parte superior da árvore
  - Há apenas um nó raiz numa árvore
- **Nó pai**
  - É o nó imediatamente acima de outro nó.
  - Todos os nós possuem um nó pai, exceto o nó raiz
- **Nó filho**
  - É um no imediatamente abaixo de outro nó.

5

Conceitos Básicos

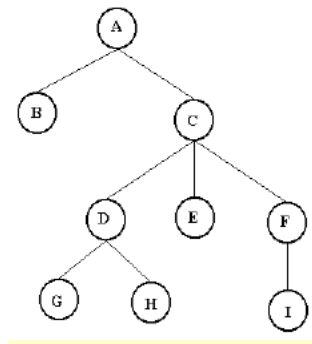


- **Grau de saída**
  - número de filhos de um nó
- **Grau de uma árvore**
  - máximo grau de saída entre todos os nós
- **Folha**
  - É o nó que não tem filhos – grau de saída nulo. Também chamado de “nó externo”
- **Nó interno**
  - É o nó que tem filho(s) – grau de saída não nulo

6

Conceitos Básicos

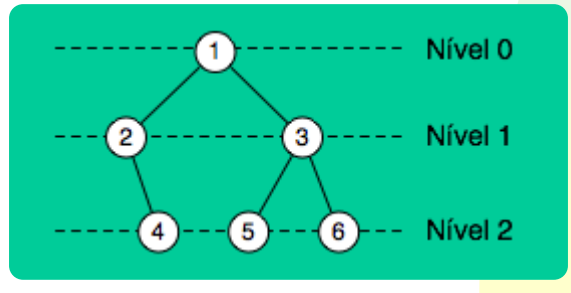
- Caminho
  - Trajeto de um nó até chegar em outro
  - Uma sequência de nós distintos v1, v2, ..., vk, tal que existe sempre entre nós consecutivos (isto é, entre v1 e v2, entre v2 e v3, ... , v(k-1) e vk) a relação “é filho de” ou “é pai de”.
- Comprimento do Caminho
  - Um caminho de vk vértices é obtido pela sequência de k-1 pares. O valor k-1 é o comprimento do caminho.



7

Conceitos Básicos

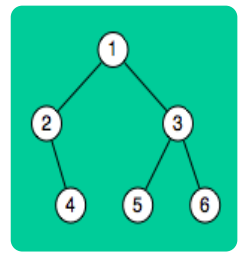
- Nível ou profundidade de um nó
  - número de nós do caminho da raiz até o nó.
  - a raiz está no nível 0, seus filhos estão no nível 1, ...
  - o último nível da árvore é a altura da árvore

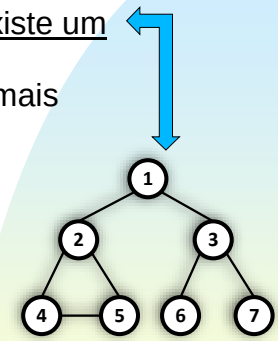


8

Conceitos Básicos

- Altura
  - propriedade fundamental das árvores: só existe um caminho da raiz para qualquer nó
  - Altura da árvore: comprimento do caminho mais longo da raiz até uma das folhas
  - a altura de uma árvore com um único nó raiz é zero
  - a altura de uma árvore vazia é -1
  - exemplo: h = 2





Não é uma árvore

9

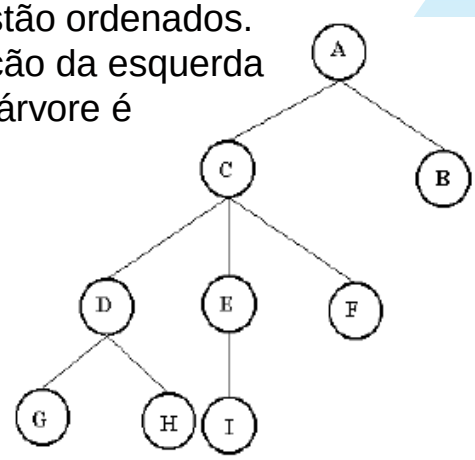
Conceitos básicos

- **Visitar**
  - Um nó é visitado quando um algoritmo atinge tal nó
- **Percorrer**
  - Percorrer uma árvore consiste em visitar todos os nós da árvore.

10

Conceitos Básicos

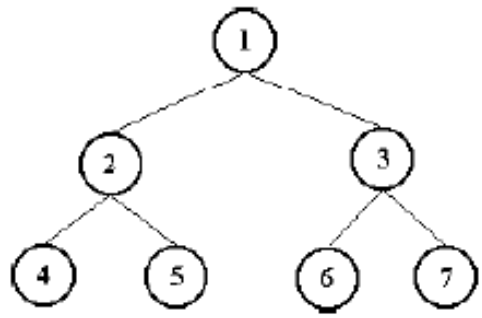
- Árvore Ordenada: é aquela na qual filhos de cada nó estão ordenados. Assume-se ordenação da esquerda para a direita. Esta árvore é ordenada?



11

Conceitos Básicos

- Árvore Cheia: Uma árvore de grau d é uma árvore cheia se possui o número máximo de nós, isto é, todos os nós têm número máximo de filhos exceto as folhas, e todas as folhas estão na mesma altura.

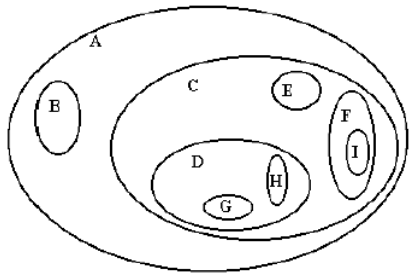


12

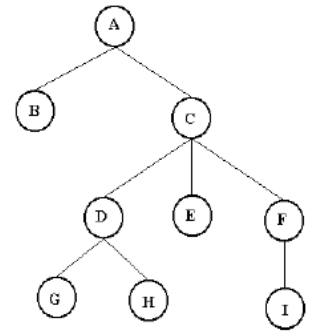
Formas de representação

- Representação por parênteses aninhados
  - (A (B) (C (D (G) (H)) (E) (F (I))))

Diagrama de Inclusão (Venn)



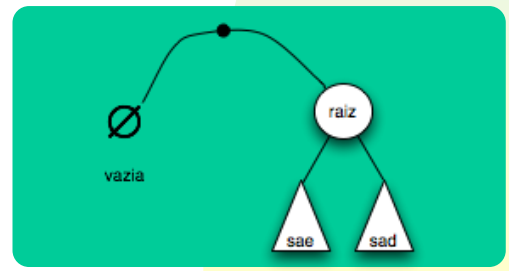
Repr. Hierárquica



13

Árvores binárias

- Cada nó tem no máximo dois filhos: 0, 1 ou 2
- Recursivamente, uma árvore binária é...
  - uma árvore vazia, ou
  - um nó raiz com duas sub-árvores
    - a sub-árvore da direita (sad)
    - a sub-árvore da esquerda (sae)

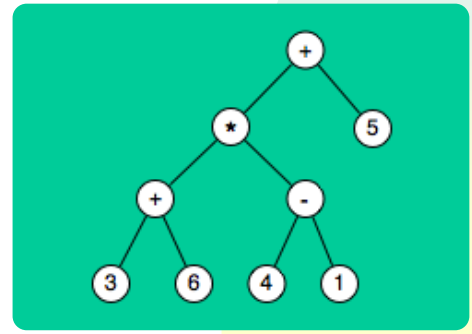


14

Árvores binárias

- Exemplo: expressões aritméticas
  - nós folhas representam operandos
  - nós internos operadores
  - exemplo:

(3+6)*(4-1)+5



15

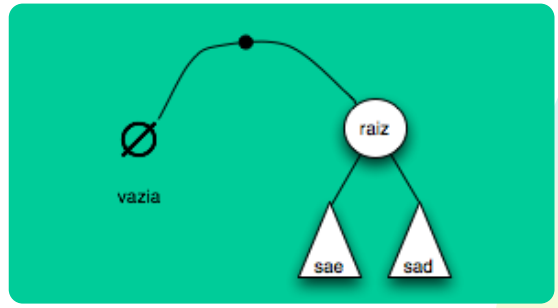
TAD Árvore Binária

- Operações:
  - Criar árvore vazia.
  - Acrescentar (enxertar) um nó ou subárvore à esquerda ou direita de um nó existente (inclusive o nó raiz).
  - Retirar (podar) um nó ou subárvore.
  - Verificar se está vazia. (*true* ou *false)*.
  - Determinar a altura da árvore.
  - Determinar o nível de um nó particular.
  - Verificar a ocorrência de um elemento particular.
  - Imprimir ou mostrar a árvore.
  - Liberar todos os recursos da árvore.

16

Implementação

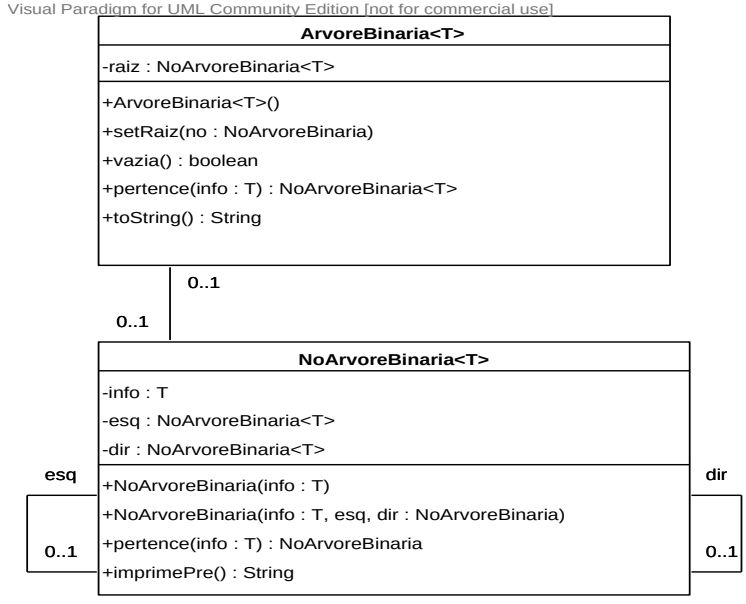
- Geralmente a implementação é recursiva
- Utiliza a definição recursiva da estrutura



- Representação da árvore: ponteiro para o nó raiz
- Representação de um nó da árvore: Classe NoArvoreBinaria

19

Implementação



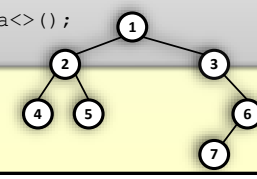
20

Inclusão de nós na árvore

- Criar objetos NoArvoreBinaria, estabelecendo a hierarquia entre eles
- Definir a raiz da árvore utilizado o método setRaiz()

Exemplo:

```
 1 NoArvoreBinaria<Integer> no4 = new NoArvoreBinaria<>(4);
 2 NoArvoreBinaria<Integer> no5 = new NoArvoreBinaria<>(5);
 3 NoArvoreBinaria<Integer> no2 = new NoArvoreBinaria<>(2, no4, no5);
 4
 5 NoArvoreBinaria<Integer> no7 = new NoArvoreBinaria<>(7);
 6 NoArvoreBinaria<Integer> no6 = new NoArvoreBinaria<>(6, no7, null);
 7
 8 NoArvoreBinaria<Integer> no3 = new NoArvoreBinaria<>(3, null, no6);
 9
10 NoArvoreBinaria<Integer> no1 = new NoArvoreBinaria<>(1, no2, no3);
11
12 ArvoreBinaria<Integer> arvore = new ArvoreBinaria<>();
13 arvore.setRaiz(no1);
```



21

Método pertence(info:T)

- Este método deve retornar um nó que contenha o valor fornecido como argumento
- Utiliza um método de mesmo nome nos objetos de NoArvoreBinaria de maneira recursiva para atingir este objetivo
- A partir da raiz da árvore, varre os nós para localizar o nó cujo conteúdo seja o argumento informado

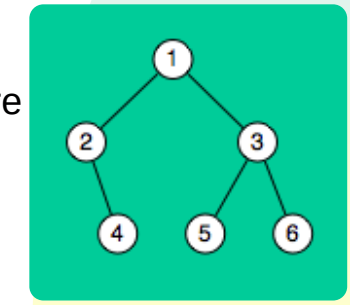
```
Algoritmo NoArvoreBinaria: pertence(info: T)

se (no.info = info) então
  retornar no;
senão
  retornar ou no.esq.pertence(info)
           ou no.dir.pertence(info);
```

22

Ordem de percurso

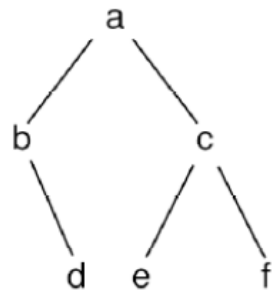
- Pré-ordem ou pré-fixada: trata raiz, percorre sae, percorre sad.
  exemplo: 1 2 4 3 5 6
- Ordem simétrica ou central: percorre sae, trata raiz, percorre sad.
  exemplo 2 4 1 5 3 6
- Pós-ordem ou pós-fixada: percorre sae, percorre sad, trata raiz.
  exemplo: 4 2 5 6 3 1



23

Árvores binárias

- Notação textual
  - árvore vazia representada por < >
  - árvores não vazias por <raiz sae sad>
  - exemplo:
    <a <b <> <d<><>> > <c <e<><>> <f<><>>> >



24

Implementação de Árvore Binária

- Lista 5

Mãos à obra !

25