Árvores n-árias

Prof. Marcel Hugo
Estruturas de Dados

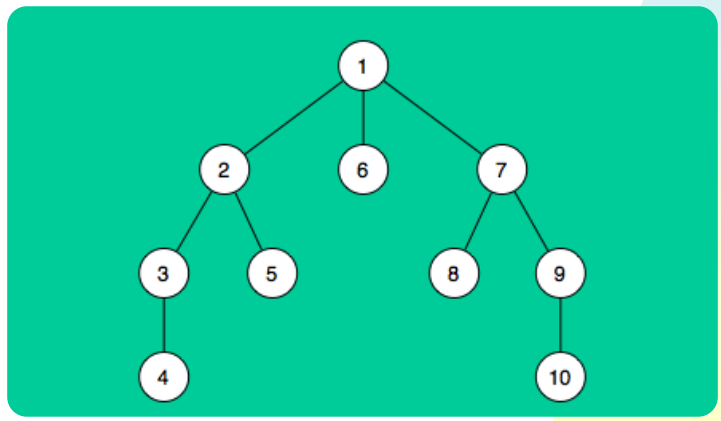
Departamento de Sistemas e Computação
Universidade Regional de Blumenau – FURB

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

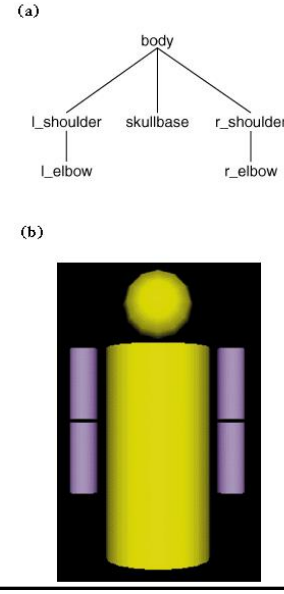
Árvores com número variável de filhos (Árvores n-árias)

- cada nó pode ter mais do que duas sub-árvores associadas
- sub-árvores de um nós são dispostas em ordem: primeira sub-árvore (sa1), segunda sub-árvore (sa2), etc...



2

Exemplos:

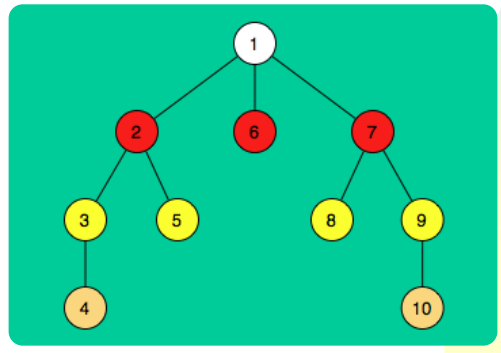


3

Árvores com número variável de filhos (Árvores n-árias)

- Notação textual: <raiz sa1 sa2 ... san>
- Exemplo:

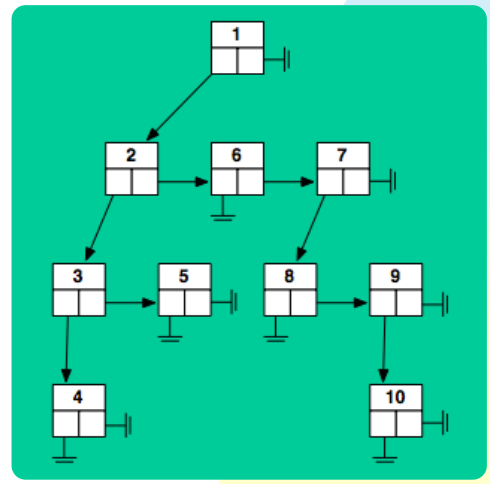
α = <1 <2 <3 <4> > <5> > <6> <7 <8> <9 <10> > > >



4

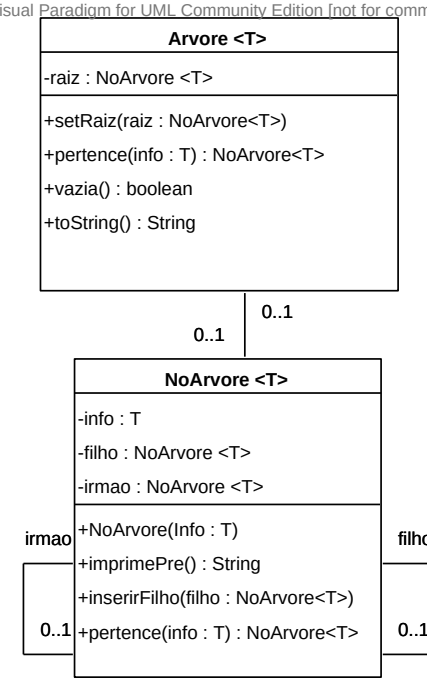
Representação (modelagem)

- Representação de árvore com número variável de filhos: utiliza uma “lista de filhos”
- um nó aponta apenas para seu primeiro filho (filho)
- cada um dos filhos aponta para o próximo irmão (irmao)



5

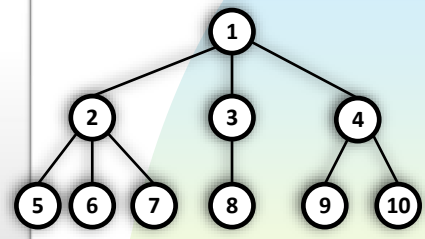
Implementação



6

Exemplo

```
NoArvore<Integer> n9 = new NoArvore<Integer>(9);
NoArvore<Integer> n10 = new NoArvore<Integer>(10);
NoArvore<Integer> n4 = new NoArvore<Integer>(4);
n4.inserirFilho(n10);
n4.inserirFilho(n9);

NoArvore<Integer> n8 = new NoArvore<Integer>(8);
NoArvore<Integer> n3 = new NoArvore<Integer>(3);
n3.inserirFilho(n8);

NoArvore<Integer> n5 = new NoArvore<Integer>(5);
NoArvore<Integer> n6 = new NoArvore<Integer>(6);
NoArvore<Integer> n7 = new NoArvore<Integer>(7);

NoArvore<Integer> n2 = new NoArvore<Integer>(2);
n2.inserirFilho(n7);
n2.inserirFilho(n6);
n2.inserirFilho(n5);

NoArvore<Integer> n1 = new NoArvore<Integer>(1);
n1.inserirFilho(n4);
n1.inserirFilho(n3);
n1.inserirFilho(n2);

Arvore<Integer> a = new Arvore<>();
a.setRaiz(n1);
```



7

Nós: criação e inserção de filho

- Cria um nó sem filhos ou "irmãos"

Algoritmo: **NoArvore(T info)**

this.info ← info;
filho ← null;
irmao ← null;

- Dado um nó já criado, definir uma sub-arvore ao nó

Algoritmo: **inserirFilho (NoArvore sa)**

sa.irmao ← filho;
filho ← sa;

8

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 NoArvore de maneira recursiva para atingir este objetivo
- O algoritmo abaixo percorre a árvore da raiz até as folhas, navegando nas sub-árvores filhas da esquerda para a direita (busca em profundidade)

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

se (no.info = info) então
  retornar no;
senão
  retornar ou no.filho.pertence(info)
           ou no.irmao.pertence(info);
```

9

Implementação de Árvore variável (genérica)

- Lista 9

Mãos à obra !

10