

Árvores

Prof. Tiago Massoni
Prof. Fernando Buarque
Engenharia da Computação
Poli - UPE

Motivação

- Para entradas realmente grandes, o acesso linear de listas é proibitivo
- Estrutura de dados não-sequencial, cujas operações tem custos em geral $O(\log n)$

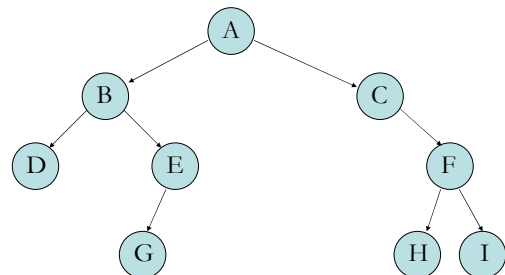
2

Árvores

- Coleção de nós em hierarquia
 - Vazia ou
 - Raíz (root) e zero ou mais subárvores (T_1, T_2, \dots, T_k) - cada uma conectada com o nó raíz
 - Raíz da subárvore é um nó filho do nó raíz
- N nós, $N-1$ arestas

3

Exemplo



4

Árvores

- Nós com mesmo pai: irmãos
- Nó sem filhos: folha
- Caminho entre os nós: sequência entre nós seguindo arestas
 - Tamanho (length): número de arestas seguidas no caminho
 - Caminho de tamanho zero: nó para ele mesmo
- Profundidade de n : tamanho do caminho único entre a raíz e o n
 - Profundidade da árvore: tamanho do maior caminho entre a raíz e uma folha
- Altura (nível) de n : tamanho do maior caminho entre uma folha e n
 - Altura de árvore = profundidade
- Se existe caminho entre dois nós: ancestral e descendente

5

Idéia geral de implementação

- Opção 1: em cada nó, uma referência para cada nó filho
 - Possível perda de espaço se não souber o número certo de filhos
- Opção 2: cada nó tem como filho uma lista ligada
 - Escalável

```
public class TreeNode <T>{  
    private <T> element;  
    private TreeNode firstChild;  
    private TreeNode nextSibling;  
}
```

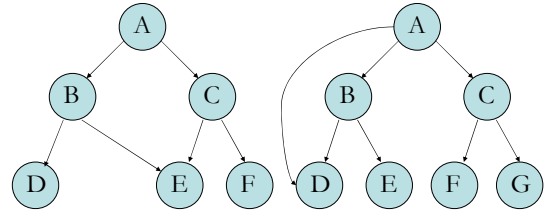
6

Árvore Binária

- Sub-árvores *esquerda* e *direita* são os sub-conjuntos formadores das árvores binárias
- Nós da árvore são todos os elementos que constituem as árvores binárias
- Profundidade da árvore é consideravelmente menor que o número de nós
- Todos os conceitos de árvores valem da mesma forma

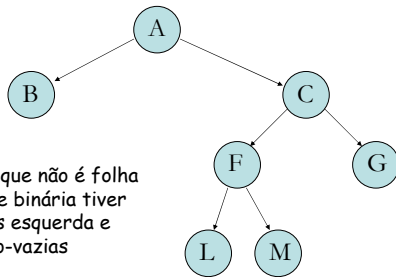
7

Intuição (Árvores não-binárias)



8

Árvores Estritamente Binárias



Se todo nó que não é folha numa árvore binária tiver sub-árvores esquerda e direitas não-vazias

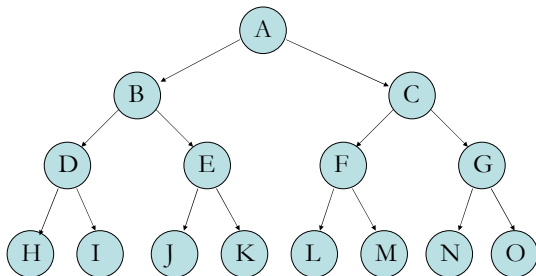
9

Árvores Binárias Completas

"Se todas as folhas de uma árvore estritamente binária de profundidade d estiverem no nível d "

10

Árvore Binária Completa Profundidade 3



11

Implementação árvores binárias

- Podemos usar referências diretas para as subárvores (left, right)

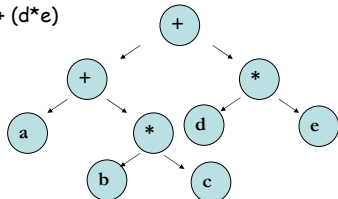
```
public class BinaryNode <T>{  
    private <T> element;  
    private BinaryNode left;  
    private BinaryNode right;  
}
```

12

Exemplo: árvores expressões

- Muito usadas em compiladores
- Folhas são operandos
- Outros nós: operadores
- Avaliação: aplica-se o operador ao resultado de avaliação de suas duas subárvores

$(a+(b*c)) + (d*e)$



13

Árvores: percursos

- Noção implícita de que árvores armazenam informações induz a necessidade de caminhar sobre a árvore
- Mas qual ordem adotar para percorrer nós?

14

Ordem para percorrer (visitar) os nós

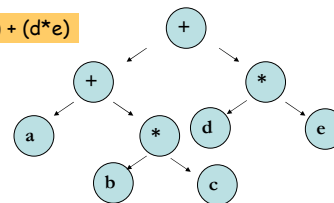
- **Pré-ordem**
 - Visita raiz
 - Percorre sub-árvore esquerda em pré-ordem
 - Percorre sub-árvore direita em pré-ordem
- **Em-ordem (simétrica)**
 - Percorre sub-árvore esquerda em ordem simétrica
 - Visita raiz
 - Percorre sub-árvore direita em ordem simétrica
- **Pós-ordem**
 - Percorre sub-árvore esquerda em pós-ordem
 - Percorre sub-árvore direita em pós-ordem
 - Visita raiz

15

No exemplo

- Em-ordem: podemos imprimir a expressão com parênteses
- Pós-ordem: notação pós-fixa
- Pré-ordem: notação pré-fixa

$(a+(b*c)) + (d*e)$



16

Opções de percursos (implementação)

- Chamadas recursivas
 - Direto: ordem das chamadas define o tipo de percurso
- Pilha auxiliar
 - Estrutura auxiliar para armazenar o caminho de retorno
- Nó father
 - Uma variável a mais em cada nó: father (null para root)

17

Outras definições

- Árvores heterogêneas
 - Não possuem o mesmo tipo nos nós (ex: expressões)
- Árvores não binárias
 - Ex: árvores B
- Árvores ordenadas
 - Ex: árvore binária de busca (veremos depois...)
- Floresta
 - Conjunto ordenado de árvores ordenadas

18

Exercício

- Algoritmo para converter uma expressão pós-fixa em uma árvore de expressões
- Dica: usa uma pilha
 - se for operando, push como árvore simples
 - se for operador, pop nas duas últimas árvores e cria uma nova árvore, com operador como raiz e as duas anteriores como sub-árvores