UNIVERSIDADE PRESBITERIANA MACKENZIE
- Faculdade de Computação e Informática –

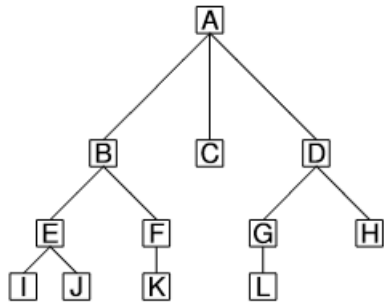
Disciplina: Estrutura de dados II
Aula 1 – Profa. Valéria Farinazzo

Tipo Abstrato de Dados (TAD) - Árvore:

Uma árvore é um tipo abstrato de dados utilizado para representar estruturas hierárquicas e as operações sobre estas estruturas de forma a se obter uma eficiência computacional adequada.

Uma árvore do tipo T é constituída de

- uma estrutura vazia, ou um elemento ou
- um nó do tipo T chamado raiz com um número finito de árvores do tipo T associadas, chamadas as sub-árvores da raiz.



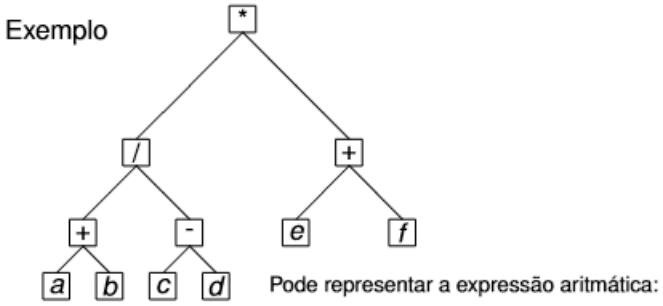
- Pai e filho: Um nó y abaixo de um nó x é chamado filho de x. x é dito pai de y. Exemplo: B é pai de E e F.
- Irmão: Nós com o mesmo pai são ditos irmãos. Exemplo: B, C, D são irmãos.
- Nível de um nó: A raiz de uma árvore tem nível 1. Se um nó tem nível i, seus filhos têm nível i +1. Exemplo: E, F, G e H têm nível 3.
- Altura ou profundidade de uma árvore: É o máximo nível de seus nós. A árvore do exemplo tem altura 4.
- Folha ou nó terminal: É um nó que não tem filhos. Exemplo: I, J, K, L são folhas.
- Nó interno ou nó não terminal: É um nó que não é folha.
- Grau de um nó: É o número de filhos do nó. Exemplo: B tem grau 2, G tem grau 1.
- Grau de uma árvore: É o máximo grau de seus nós. A árvore do exemplo tem grau 3.

Árvores Binárias

Árvore binária: É uma árvore ordenada de grau 2.

Uma árvore binária é

- vazia ou
- um nó chamado raiz mais duas árvores binárias disjuntas chamadas subárvore esquerda e subárvore direita.



((a +b) /(c−d)) ∗(e + f)
Veja como a estrutura de árvore binária expressa de maneira clara a precedência das operações, sem necessidade de usar parêntesis.

Aplicações que utilizam árvores ou árvores binárias

- Problemas de busca de dados armazenados na memória principal do computador: árvore binária de busca, árvores (quase) balanceadas como AVL, rubro-negra, etc.
- Problemas de busca de dados armazenados na memória secundárias principal do computador (disco rígico): e.g. B-árvores.
- Aplicações em Inteligência Artificial: árvores que representam o espaço de soluções, e.g. jogo de xadrez, resolução de problemas, etc.
- No processamento de cadeias de caracteres: árvore de sufixos.
- Na gramática formal: árvore de análise sintática.
- Em problemas onde a meta é achar uma ordem que satisfaz certas restrições (e.g. testar a propriedade de uns-consecutivos numa matriz, reconhecer grafos intervalo, planaridade de grafo, etc.): árvore-PQ.

Formas de Percorrer uma Árvore

Em algumas aplicações, é necessário percorrer uma árvore de forma sistemática, visitando cada nó da árvore uma única vez, em determinada ordem.
Por exemplo, se cada nó da árvore possui um campo que armazena o salário, então podemos querer visitar cada nó para fazer um reajuste salarial. A visita seria atualizar o campo salário.
Não podemos esquecer nenhum nó, nem queremos visitar um nó mais do que uma vez. Neste caso, a ordem de visita não é importante.
Mas em algumas outras aplicações, queremos visitar os nós em certa ordem desejada. Veremos várias formas para percorrer uma árvore binária.
Pré-ordem.
In-ordem ou ordem simétrica.

Pós-ordem.

Percorrendo uma árvore binária em **pré-ordem**: ao percorrer a árvore em pré-ordem, primeiro visita-se o nó em questão para depois visitar a árvore em pré-ordem começando pelo filho da esquerda e depois visitar a árvore em pré-ordem começando pelo filho da direita, recursivamente:

a. Visita o nó considerado.
b. Percorre em pré-ordem nó→filho da esquerda.
c. Percorre em pré-ordem nó→filho da direita.

Uma segunda forma de se percorrer uma árvore binária chama-se **em-ordem**: ao percorrer a árvore em-ordem, primeiro percorremos a sub-árvore da esquerda, depois visitamos o nó sendo considerado para depois percorrer a sub-árvore da direita.

a. Percorre em-ordem nó→filho da esquerda.
b. Visita o nó considerado.
c. Percorre em-ordem nó→filho da direita.

A terceira forma de se percorrer uma árvore binária chama-se em **pós-ordem**: ao percorrer a árvore em pós-ordem, percorremos primeiro os filhos da esquerda e da direita, nesta ordem, para depois visitar o nó considerado.

a. Percorre em pós-ordem nó→filho da esquerda.
b. Percorre em pós-ordem nó→filho da direita.
c. Visita o nó considerado.