



Árvores

Uma das mais importantes classes de estruturas de dados em computação são as árvores. Aproveitando-se de sua organização hierárquica, muitas aplicações são realizadas usando-se algoritmos relativamente simples, recursivos e de eficiência bastante razoável.

Definições e representações básicas

Uma árvore é uma estrutura de dados que se caracteriza por uma relação de hierarquia entre os elementos que a compõem. Exemplos de estruturas em forma de árvores são:

- O organograma de uma empresa;
- A divisão de um livro em capítulos, seções, tópicos, etc;
- A árvore genealógica de uma pessoa.

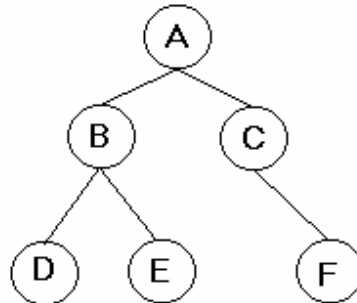
De um modo um pouco mais formal, podemos dizer que uma árvore é um conjunto finito de um ou mais nodos (nós ou vértices), tais que:

1. existe um nodo denominado **raiz**;
2. os demais nodos formam $m \geq 0$ conjuntos disjuntos s_1, s_2, \dots, s_m , tais que cada um desses conjuntos também é uma árvore (denominada sub-árvore).

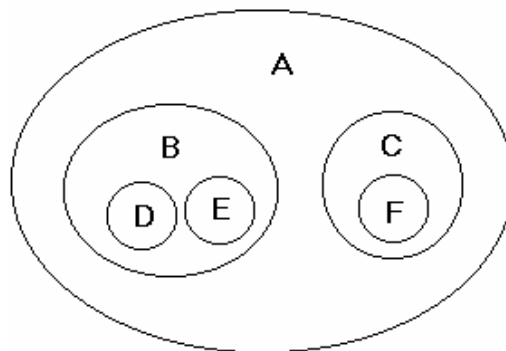
Uma floresta é um conjunto de árvores. Se v é um nodo de A , a notação A_v indica a subárvore de v com raiz A .

Para visualizar esse conceito, pode-se representá-lo graficamente. Há formas diferentes de representações gráficas de uma árvore. Em todas elas, cada nodo poderá ser associado a um identificador, denominado *rótulo*.

a) Representação hierárquica



b) Representação por conjuntos (diagrama de inclusão)



c) Representação por expressão parentetizada (parênteses aninhados)

Cada conjunto de parênteses correspondentes contém um nodo e seus filhos. Se um nodo não tem filhos, ele é seguido por um par de parênteses sem conteúdo.

(A (B (D () E ())) (C (F ()))))

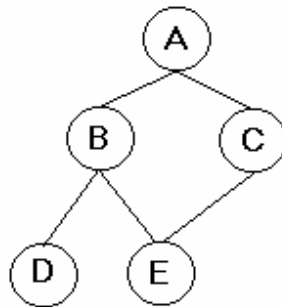
d) Representação por expressão não parentetizada

Cada nó é seguido por um número que indica a quantidade de filhos desse nodo, e em seguida por esses filhos, representados do mesmo modo.

A 2 B 2 D 0 E 0 C 1 F 0

Pode-se representar uma árvore de muitos outros modos, mas é interessante notar que, dentre os exemplos apresentados, a representação **a)** é a que permite uma melhor visualização, e que será utilizada a partir deste ponto. As representações **c)** e **d)** não permitem boa visualização da estrutura, mas podem ser úteis para guardar em arquivos os dados de uma árvore.

Como, por definição, os subconjuntos s_1, s_2, \dots, s_m são disjuntos, cada nó só pode ter um pai. Assim, o desenho abaixo, por exemplo, não representa uma árvore:



Definições

Dada uma árvore qualquer:

- 1) A linha que liga dois nodos da árvore denomina-se aresta.
- 2) Diz-se que existe caminho entre dois nodos V e W da árvore, se a partir do nodo V puder-se chegar ao nodo W percorrendo-se as arestas que ligam os nodos intermediários entre V e W.

Observa-se que existe sempre um caminho entre a raiz e qualquer nodo da árvore.

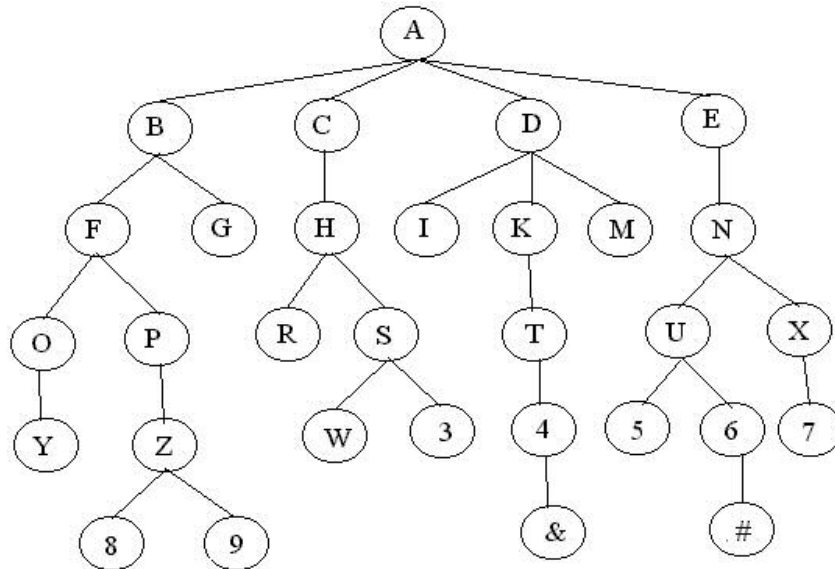
- 3) Se houver um caminho entre V e W, começando em V diz-se que V é um nodo ancestral de W e W é um nodo descendente de V. Se este caminho contiver uma única aresta, diz-se que V é o nodo pai de W e que W é um nodo filho de V. Dois nodos que são nodos filhos do mesmo nodo pai são denominados nodos irmãos.

Uma característica inerente a árvores é que qualquer nodo, exceto a raiz, tem um único nodo pai.

- 4) Se um nodo não possui nodos descendentes, ele é chamado de folha ou nodo terminal da árvore.
- 5) Grau de um nodo é o número de nodos filhos do mesmo. Obviamente que um nodo folha tem grau zero.
- 6) Nível de um nodo é o número de nodos existentes no caminho entre a raiz e o próprio nodo. A raiz tem nível 1.
- 7) O grau da árvore é igual ao grau do nodo de maior grau da árvore.

8) O nível da árvore é igual ao nível do nodo de maior nível da árvore.

Exercício:



Responda:

- Qual é a raiz da árvore?
- Quais são os nodos terminais?
- Qual o grau da árvore?
- Qual o nível da árvore?
- Quais são os nodos descendentes do nodo **D** ?
- Quais são os nodos ancestrais do nodo # ?

g) Os nodos 4 e 5 são nodos irmãos?

h) Há caminho entre os nodos C e S?

i) Qual o nível do nodo 5?

j) Qual o grau do nodo A?

Árvores Binárias

Conforme já mencionado, as árvores constituem as estruturas não seqüenciais com maior aplicação em computação. Dentre as árvores, as binárias são, sem dúvida, as mais comuns.

Deve-se observar que a ordem em que estão posicionadas as subárvores em relação à raiz é fundamental. Uma vez que cada nodo tem no máximo dois nodos filhos, cada um destes nodos (se houverem) são identificados segundo a sua posição relativa à raiz. Distingue-se, então, o nodo filho à esquerda do nodo filho à direita e, conseqüentemente, a subárvore à esquerda da subárvore à direita.

Exemplo:

Nos exemplos acima, as árvores apresentadas são distintas, pois no primeiro exemplo, o nodo A tem somente um filho à esquerda e no segundo exemplo, o nodo tem um filho à direita

Transformação em árvore binária

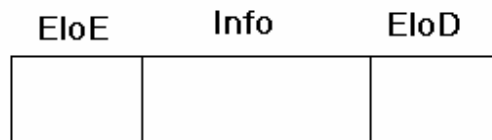
- 1) a raiz da árvore (subárvore) será a raiz da árvore (subárvore) binária.
- 2) O nodo filho mais à esquerda da raiz da árvore (subárvore) será o nodo filho à esquerda da raiz da árvore (subárvore) binária. Cada nodo irmão de V, da esquerda para a

direita, será o nodo filho à direita do nodo irmão da esquerda, até que todos os nodos filhos da raiz da árvore (subárvore) já tenham sido incluídos na árvore binária em construção.

Exemplo: (converta a árvore anteriormente apresentada em uma árvore binária)

Representação de árvores binárias (implementação)

Cada nodo componente de uma árvore binária tem no máximo dois filhos, sendo necessários, portanto, dois campos de elos para representa-lo. A configuração de um nodo numa árvore binária é a que segue:



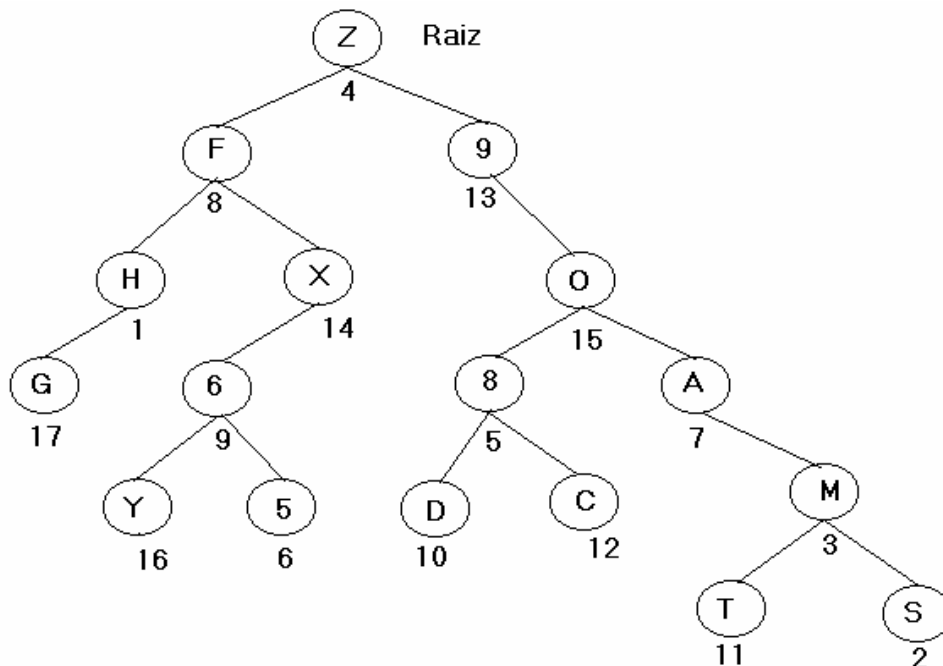
Sendo que EloE (i) indica o endereço do nodo filho à esquerda do nodo de índice i e EloD (i) aponta para o nodo filho à direita. Uma árvore binária pode, então, ser representada por;

- uma matriz Info que conterà os campos de informações dos nodos, e
- dois vetores de ligação, EloE e EloD.

ou uma estrutura com um campo info, um campo EloE e um campo EloD.

Sendo conhecido o índice do nodo raiz de uma árvore, a mesma é completamente definida (supõe-se que o índice da raiz esteja contido na variável RAIZ).

Exemplo: a seguinte árvore (os números abaixo de cada nodo indicam o índice do mesmo)



Pode ser representada por:

Índice	EloE	Info	EloD
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			

Caminhamento em árvores binárias

Há diversas formas de manipulação de árvores binárias e, em geral, estas supõem o exame dos conteúdos (informações) dos nodos. O acesso sistemático aos nodos de uma árvore de maneira que cada nodo seja examinado no máximo uma única vez (para que não haja repetição), sugere que a árvore seja percorrida segundo algum critério pré-estabelecido. Esta ação de percorrer a árvore, com a condição de que cada nodo seja examinado no máximo uma vez, denomina-se caminhamento na árvore binária.

Se for realizado um caminhamento numa árvore de forma que todos os seus nodos sejam visitados (acessados), os mesmos são implicitamente organizados segundo uma ordem linear. Dependendo do critério estabelecido para caminhar na árvore, obtém-se uma seqüência dos nodos correspondentes $x_1 x_2 \dots x_n$, em que:

- a) n é o número de nodos da árvore;

- b) x_j é o conteúdo do nodo que ocorre na j -ésima posição na seqüência de caminhamento da árvore; e
- c) x_i ocorre antes de x_j na seqüência se o nodo com informação x_i é visitado antes do que o nodo x_j , segundo o caminhamento escolhido.

Em geral são utilizadas três formas de caminhamentos em árvores binárias e estas são determinadas dependendo da ordem em que são visitados o nodo raiz, sua subárvore à esquerda e sua subárvore à direita (o termo “visitar” significa a realização de alguma operação sobre a informação do nodo, como modificação da mesma, impressão ou qualquer outra). Os três caminhamentos usuais são descritos como segue:

- Caminhamento **PRÉ_FIXADO** (raiz-esquerda-direita)

- a) visitar a raiz;
- b) caminhar na subárvore à esquerda, segundo este caminhamento;
- c) caminhar na subárvore à direita, segundo este caminhamento.

Exemplo:

- Caminhamento **INFIXADO** (esquerda-raiz-direita)

- a) caminhar na subárvore à esquerda, segundo este caminhamento;
- b) visitar a raiz;
- c) caminhar na subárvore à direita, segundo este caminhamento.

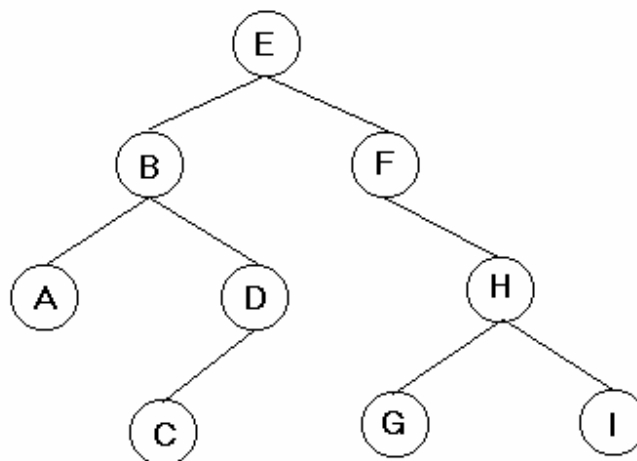
Exemplo:

- Caminhamento **PÓS-FIXADO** (esquerda-direita-raiz)
 - a) caminhar na subárvore à esquerda, segundo este caminhamento
 - b) caminhar na subárvore à direita, segundo este caminhamento
 - c) visitar a raiz;

Exemplo:

Árvores Binárias de Pesquisa

Árvores binárias de pesquisa são aquelas nas quais os dados são distribuídos pelos nós de forma a facilitar a pesquisa de um determinado elemento. A distribuição dos dados é feita de tal modo que, em qualquer subárvore, todos os elementos situados à esquerda da raiz são menores que o elemento da raiz, e todos os situados à direita da raiz são maiores que o da raiz. A figura abaixo mostra um exemplo de uma árvore binária de pesquisa.

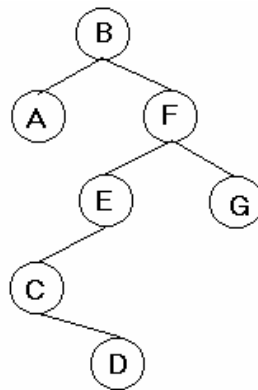


Com essa distribuição, a pesquisa de um elemento qualquer fica mais eficiente. Em qualquer nível, verifica-se primeiramente a raiz. Se o dado procurado estiver na raiz, a

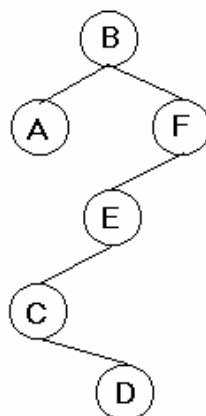
pesquisa está encerrada. Caso contrário, se o elemento for menor que o da raiz, procura-se na subárvore esquerda; se for maior, procura-se na subárvore da direita. Mas nunca é necessário procurar nas duas subárvores.

A inserção é feita mantendo a disposição dos elementos de forma que a árvore continue a ser binária de pesquisa. É feita inicialmente uma procura pelo 'novo elemento' na árvore: se ele já estiver presente não inclui-se ele; caso contrário é criado e inserido o 'novo elemento' na posição onde ele deveria estar presente.

A remoção de um nodo de uma árvore binária de pesquisa deve ser efetuada com cuidado, para que os nodos que permanecem na árvore continuem distribuídos de acordo com a mesma estratégia que permite a pesquisa binária. Para isso, devem ser analisados três casos, usando como exemplo a figura abaixo.

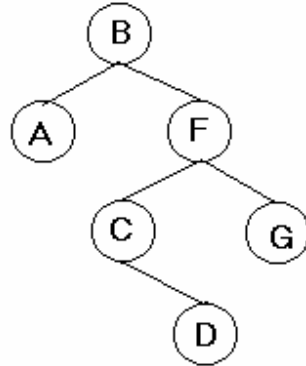


1. O nodo a ser retirado é uma folha, isto é, não tem filhos. Neste caso o nodo é simplesmente retirado, como por exemplo o nodo G do exemplo. Depois da remoção, a árvore resultante fica da seguinte maneira:



2. O nodo a ser retirado tem apenas um filho. Neste caso o nodo é retirado e em seu lugar é colocada a raiz da sua única subárvore. No exemplo, o nodo retirado é o E,

cujo lugar vai ser ocupado pelo nodo C. Depois da remoção, a árvore resultante fica como a figura abaixo:



3. O nodo a ser retirado tem dois filhos. Neste caso, não se retira o nodo desejado, mas altera-se o seu conteúdo. Por exemplo, se o nodo desejado é o B, altera-se o seu conteúdo para o nodo seguinte na ordem, que é o C, e em seguida remove-se o nodo que continha originalmente o valor C. Para determinar o nodo seguinte a outro, caminha-se para a subárvore da direita do nodo original e, em seguida desce-se sempre à esquerda, até encontrar o nodo que não tenha filhos à esquerda. Depois da remoção, a árvore fica como a figura abaixo:

