ÁRVORE

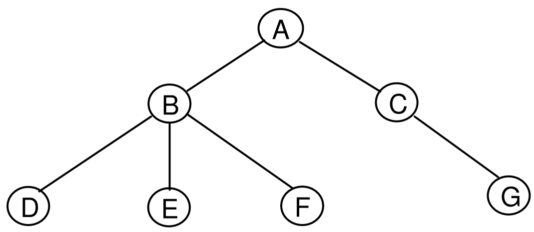
Provavelmente a estrutura não linear de maior aplicação em computação é a chamada estrutura de *árvore*, ou simplesmente árvore, que pode ser definida da seguinte maneira.

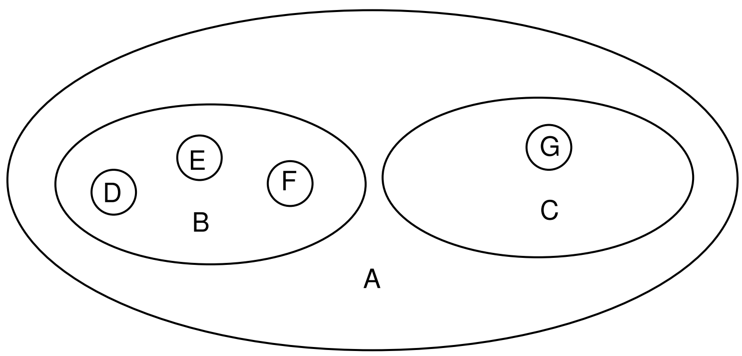
Uma *árvore* A é um conjunto finito de N nodos, tal que se N > 0, então

i) existe um nodo especial denominado *raiz*;
ii) os restantes N - 1 nodos estão particionados em M conjuntos disjuntos, $A_1, \ldots, A_M$, cada um dos quais é por sua vez uma árvore. As árvores $A_1, \ldots, A_M$ são denominadas *subárvores* da raiz.

Esta é uma definição recursiva. A recursividade é aqui utilizada como uma ferramenta para a definição de um padrão fundamental de estruturação, permitindo que se defina de uma maneira elegante e concisa estruturas com um grau maior de sofisticação, muito além da capacidade da repetição ou do seqüenciamento.

Existem diversas possibilidades para realizar a representação gráfica de uma árvore. Abaixo são ilustradas algumas das possíveis formas de representar graficamente a árvore { A, B, C, D, E, F, G }. Nesta árvore, A é a raiz e { B, D, E, F } e { C, G } são subárvores de A. { D }, { E } e { F } são subárvores da raiz B e { G } é subárvore da raiz C.



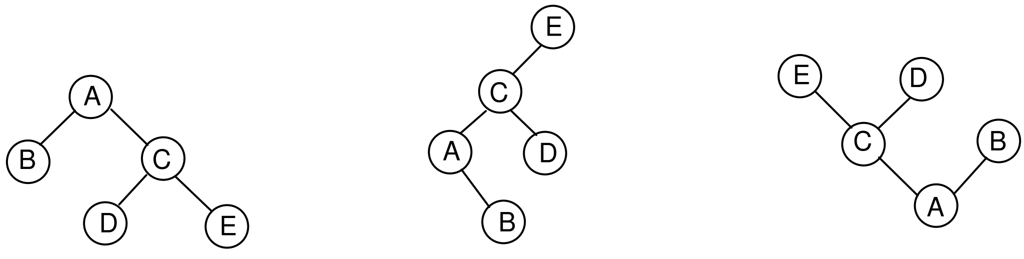


(A (B (D, E, F), C (G)))

A
B
D
E
F
C
G

A representação de árvores em forma de grafo, apesar de ser a que mais justifica o nome dado à estrutura, está colocada de uma maneira bastante incomum entre as árvores da natureza - com a raiz em cima. Na verdade este é um costume generalizado em computação e, provavelmente, teve sua origem na representação de estruturas hierárquicas por árvores.

Embora as três representações da figura abaixo serem formas válidas de se representar graficamente uma árvore, existe uma terminologia associada com a disposição gráfica dos nodos que já se tornou padrão.



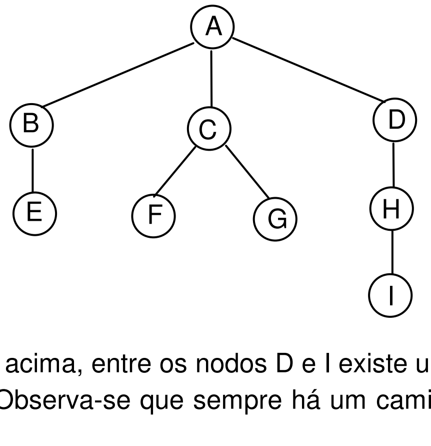
a) b) c)

Diz-se, por exemplo, que B e C estão abaixo de A, C está acima de D e E ou então que B é o nodo mais à esquerda da árvore. Assim, somente a representação (a) está coerente com esta terminologia, onde tem-se a raiz no nível superior do grafo e as subárvores da raiz se desenvolvendo de cima para baixo.

A árvore representada em (a) tem como raiz o nodo com informação A e duas subárvores. Os nodos B, D e E são raízes cujas subárvores respectivas são vazias.

Dada uma árvore qualquer representada através de um grafo :

i) a linha que liga dois nodos da árvore denomina-se *aresta*;
ii) diz-se que existe um *caminho* entre dois nodos *v* e *w* da árvore, se a partir do nodo *v* for possível alcançar o nodo *w* percorrendo-se as arestas que ligam os nodos *intermediários* entre *v* e *w*;



Considerando a árvore acima, entre os nodos D e I existe um caminho, entretanto não existe caminho entre B e G. Observa-se que sempre há um caminho entre a raiz e qualquer nodo da estrutura de árvore.

iii) 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 *v* é o nodo *pai* de *w* e *w* é um nodo *filho* de *v*. Dois nodos filhos do mesmo nodo são denominados nodos *irmãos*;

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

Considerando a estrutura de árvore anterior, tem-se as seguintes relações, entre outras :
a) o nodo D é um nodo ancestral de I;
b) G é filho de C;
c) B é pai de E;
d) H é descendente de D;
e) A é ancestral de todos os nodos da árvore;
f) os nodos F e G são irmãos.

iv) se um nodo não possui nodos descendentes ele é chamado de nodo *terminal* ou *folha* da árvore;

E, F, G e I são nodos terminais ou folhas da árvore acima representada.

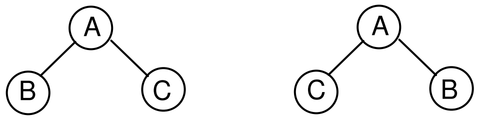
v) *grau* de um nodo é o número de subárvores que possui ou o número de filhos do mesmo. Um nodo folha tem grau 0;

A tem grau 3, C tem grau 2, D tem grau 1 e G tem grau 0.

vi) *nível* de um nodo corresponde ao número de nodos existentes no caminho entre a raiz e o próprio nodo ou, ainda, se um determinado nodo está localizado no nível *i*, então as raízes de suas subárvores estão no nível *i + 1*. Arbitrariamente o nível da raiz da árvore é 1;

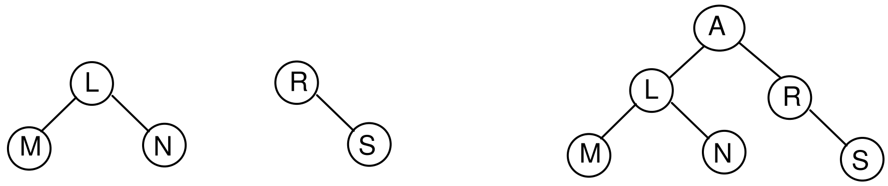
Os níveis de E, C, A e I são 3, 2, 1 e 4, respectivamente.

Uma árvore é *ordenada* quando a ordem das subárvores $A_1, \ldots, A_M$ na definição é importante; nesse caso A_1 é a primeira subárvore da raiz, A_2 a segunda e assim sucessivamente, tantas quantas forem as subárvores. Quando a ordem das subárvores é irrelevante, diz-se que a árvore é *orientada*, no sentido de que as árvores abaixo representadas são equivalentes, o que não ocorre com árvores ordenadas.



As árvores ordenadas ocorrem com maior freqüência em computação e por isso são simplesmente denominadas árvores.

Uma *floresta* é um conjunto de zero ou mais árvores disjuntas. A distinção entre florestas e árvores é mínima : basta acrescentar um nodo a uma floresta para que ela se transforme em uma árvore, enquanto que toda árvore é uma floresta.

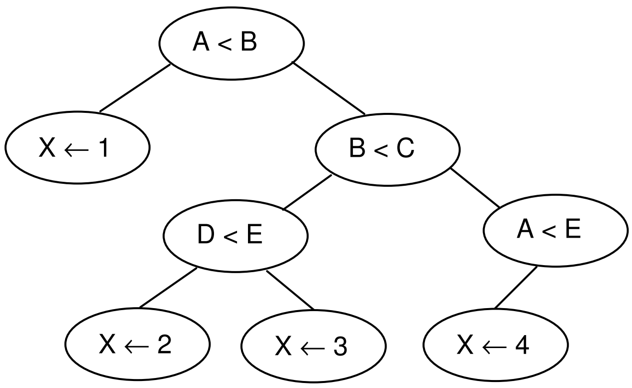


ÁRVORES BINÁRIAS

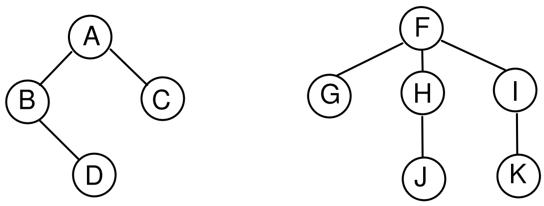
Uma árvore binária é uma árvore ordenada na qual cada nodo possui no máximo duas subárvores, distinguidas entre *subárvore esquerda* e *subárvore direita*. Quando um nodo tem apenas uma subárvore, ela será a subárvore esquerda ou a subárvore direita, exclusivamente. Desta forma, as árvores binárias abaixo representadas são diferentes.

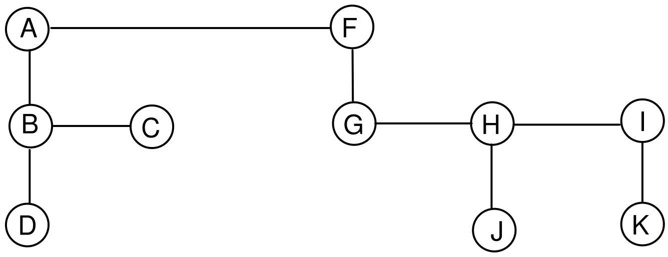
Árvores genealógicas são exemplos de árvores binárias. Um ninho de Ses em programação também pode ser representado através de uma estrutura de árvore binária, onde cada nodo da subárvore à esquerda corresponde à cláusula *então* e os nodos da subárvore à direita à cláusula *senão.*

```
INÍCIO
   SE  A  <  B  ENTÃO
      X  ←  1
   SENÃO
      SE  B  <  C  ENTÃO
         SE  D  <  E  ENTÃO
            X  ←  2
         SENÃO
            X  ←  3
         FIM SE
      SENÃO
         SE  A  <  E  ENTÃO
            X  ←  4
         FIM SE
      FIM SE
   FIM SE
FIM
```

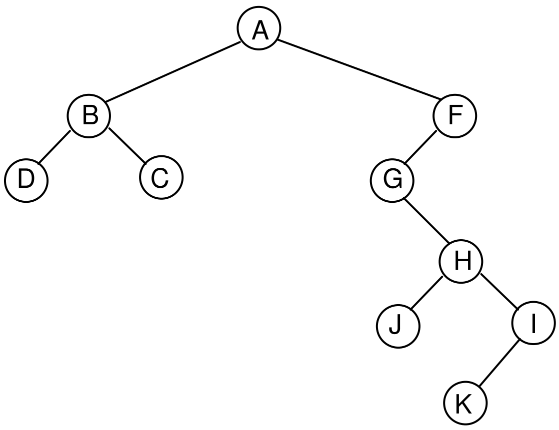


A ocorrência de árvores binárias em computação é suficientemente grande para justificar o seu estudo mais detalhado. Além disso, é possível representar qualquer floresta como uma árvore binária. Por exemplo, a floresta formada pelas duas árvores imediatamente abaixo pode ser representada pela árvore binária que segue.





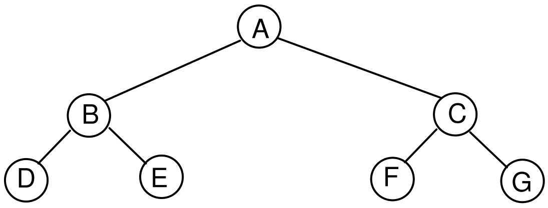
Esta árvore binária foi obtida a partir da floresta anterior unindo todos os filhos de uma mesma família (pelo lado direito) e removendo todas as ligações de pai para filho com exceção do primogênito. Girando a figura a 45º obtém-se a árvore binária no esquema tradicional.

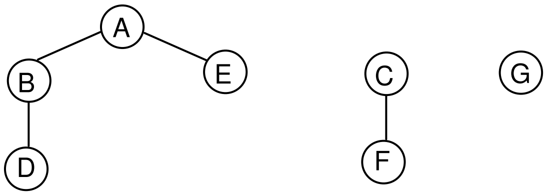


A transformação de uma árvore qualquer para uma árvore binária pode ser feita como segue.

i) a raiz da árvore (subárvore) será a raiz da árvore (subárvore);
ii) 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, 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.

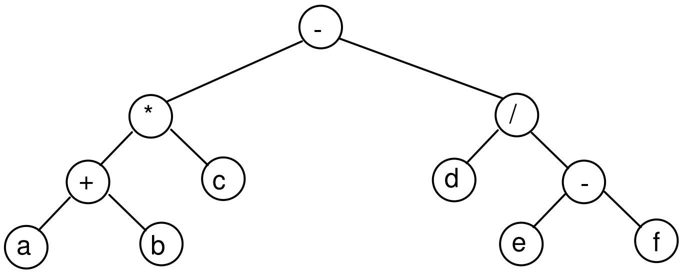
Invertendo o processo, a árvore binária abaixo corresponde à floresta seguinte.





As transformações anteriores formam a chamada correspondência natural entre árvores binárias e florestas.

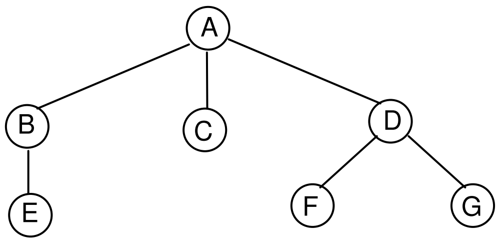
Um exemplo interessante de árvore binária é uma expressão aritmética com operadores diádicos (aplicados sobre dois operandos), onde cada operador possui como subárvores os seus operandos. A expressão (a + b) * c - d / (e - f) representada através de uma árvore binária corresponderia a



REPRESENTAÇÃO DE ÁRVORES

Alocação estática

1ª Representação : vetor bidimensional, onde são armazenados a informação e o grau de cada nodo da estrutura de árvore.



INFORMAÇÃO	GRAU
A	3
B	1
C	0
D	2
E	0
F	0
G	0

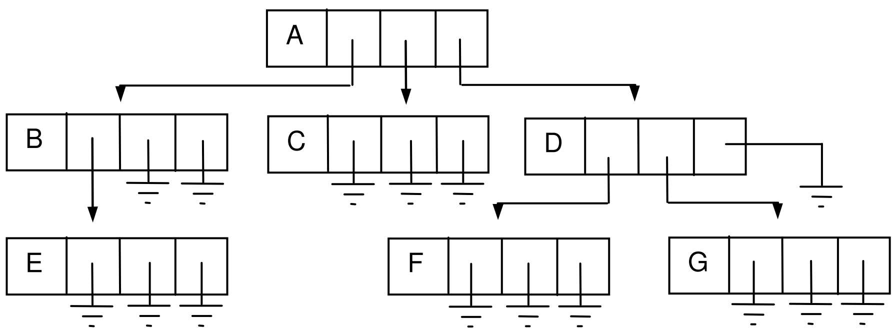
2ª Representação : vetor bidimensional, onde são armazenadas a informação e a(s) raiz(ízes) da(s) subárvore(s) da árvore.

	INFORMAÇÃO	RAIZ(ÍZES) DA(S) SUBÁRVORE(S)		
1	A	2	3	4
2	B	5	-1	-1
3	C	-1	-1	-1
4	D	6	7	-1
5	E	-1	-1	-1
6	F	-1	-1	-1
7	G	-1	-1	-1

Nesta forma de representação, o número de colunas do vetor bidimensional deve corresponder ao grau máximo da árvore + 1.

Alocação dinâmica

Na representação de uma estrutura de árvore, quando alocada dinamicamente em memória, a quantidade de campos necessários para indicar a(s) raiz(ízes) da(s) subárvore(s) do nodo representado é determinada como sendo o maior grau entre os graus da coleção de nodos da árvore, pois todos os nodos pertencentes a uma estrutura devem ter a mesma configuração.

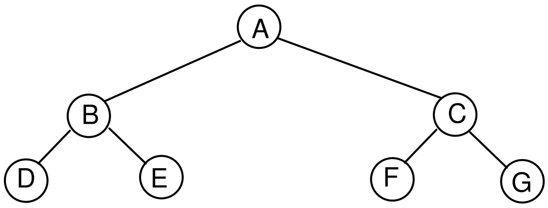


REPRESENTAÇÃO DE ÁRVORES BINÁRIAS

Alocação estática

1ª Representação

Cada nodo componente de uma árvore binária tem no máximo dois filhos. Desta forma, a alocação estática pode utilizar um vetor bidimensional como forma de representação, onde são armazenadas a informação e as raízes das subárvores esquerda e direita.



	RAIZ SUBÁRVORE ESQUERDA	INFORMAÇÃO	RAIZ SUBÁRVORE DIREITA
1	2	A	3
2	4	B	5
3	6	C	7
4	-1	D	-1
5	-1	E	-1
6	-1	F	-1
7	-1	G	-1

Nesta forma de representação, o número de colunas do vetor bidimensional é três.

2ª Representação

As informações dos nodos de uma árvore binária podem ser armazenadas em um vetor unidimensional, sendo que a informação do nodo com numeração *i* será armazenada na i-ésima posição do vetor.

A árvore binária anterior, nesta forma de representação, seria estruturada como segue.

1	2	3	4	5	6	7
A	B	C	D	E	F	G

Para qualquer nodo representado na posição *i* do referido vetor tem-se :

i) o *pai* de *i* está em *i* DIV 2, sendo $i \neq 1$; se $i = 1$, *i* é a raiz da árvore e não possui pai;

ii) o *filho esquerdo* de *i* está em $2i$, se $2i \leq n$; se $2i > n$, *i* não tem filho esquerdo;

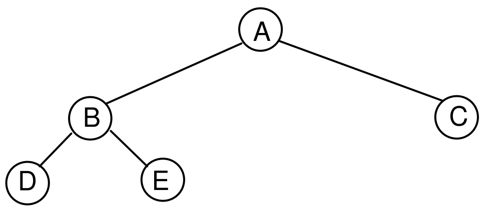
iii) o *filho direito* de *i* está em $2i + 1$, se $2i + 1 \leq n$; se $2i + 1 > n$, *i* não tem filho direito.

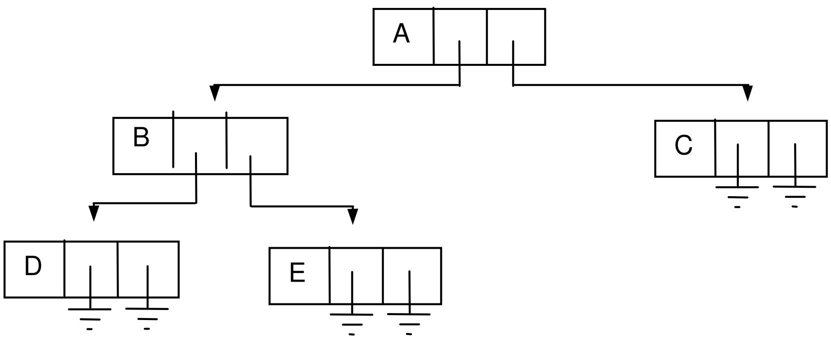
Essa representação pode ser utilizada para todas as árvores binárias, embora na maioria dos casos haverá espaço não utilizado. Para árvores binárias *completas* a representação é ideal, uma vez que não há desperdício de posições de memória.

Alocação dinâmica

Embora a representação acima pareça satisfatória para as árvores binárias completas, desperdiça espaço em memória no caso de muitas outras árvores binárias. A representação sofre, além do mais, das inadequações gerais das representações seqüenciais. A inserção ou eliminação dos nodos do meio de uma árvore requer o movimento em potencial de muitos nodos para refletir a mudança no nível desses nodos. Esses problemas podem facilmente ser superados mediante a adoção de uma representação através de alocação dinâmica.

Nesta forma de representação são necessários dois campos para indicar as raízes das subárvores esquerda e direita, além de um campo para armazenar a informação do nodo.





CAMINHAMENTO EM ÁRVORES BINÁRIAS

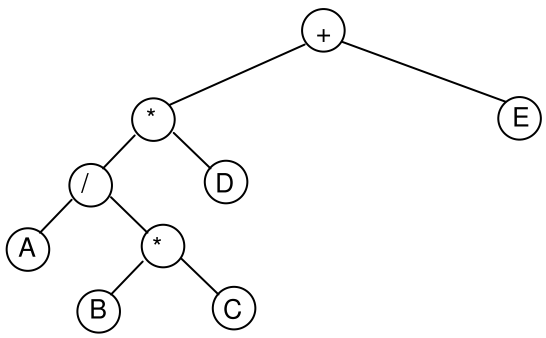
Um problema que surge em muitos algoritmos que manipulam árvores binárias é o de percorrer todos os seus nodos para examinar ou modificar seu conteúdo, de tal maneira que cada um dos nodos seja visitado exatamente uma vez. Esta ação de percorrer a árvore com a condição de que cada nodo seja examinado uma e apenas uma única vez é denominado *caminhamento em árvore binária*.

Um caminhamento completo através da árvore determina uma seqüência de nodos; como em muitos problemas é necessário conhecer o nodo subseqüente (ou anterior) nesta seqüência, é conveniente definir uma sistemática para o caminhamento.

Em geral são utilizadas três formas de caminhamento em árvores binárias e estas são determinadas dependendo da ordem em que forem visitados o nodo raiz, a subárvore esquerda e a subárvore direita (o termo *visitar* significa a realização de alguma operação sobre a informação do nodo).

Os três caminhamentos usuais em árvores binárias são descritos como segue :

a) caminhamento em *pré-ordem*
   i) visite a raiz;
   ii) caminhe através da subárvore esquerda em pré-ordem, se existir;
   iii) caminhe através da subárvore direita em pré-ordem, se existir.



+ * / A * B C D E

b) caminhamento *inordem*
   i) caminhe através da subárvore esquerda em inordem, se existir;
   ii) visite a raiz;
   iii) caminhe através da subárvore direita em inordem, se existir.

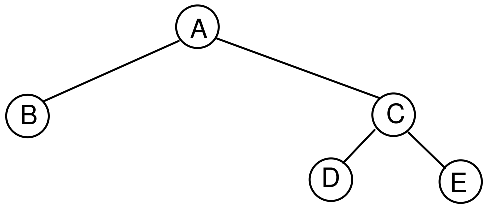
A / B * C * D + E

c) caminhamento em *pós-ordem*
   i) caminhe através da subárvore esquerda em pós-ordem, se existir;
   ii) caminhe através da subárvore direita em pós-ordem, se existir;
   iii) visite a raiz.

A B C * / D * E +

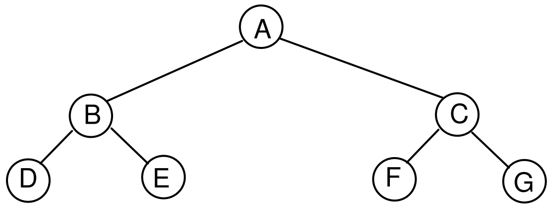
ÁRVORE BINÁRIA COMPLETA

Uma *árvore binária completa* é aquela que apresenta a seguinte propriedade : se *v* é um nodo tal que alguma subárvore de *v* é vazia então *v* se localiza ou no último ou no penúltimo nível da árvore.



ÁRVORE BINÁRIA CHEIA

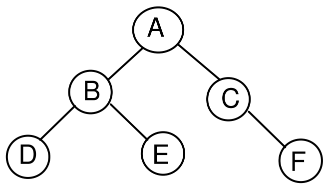
Uma *árvore binária cheia* é aquela em que, se *v* é um nodo com alguma de suas subárvores vazias entào *v* se localiza no último nível da árvore.



ÁRVORE BINÁRIA BALANCEADA

Sucessivas inserções e retiradas em uma árvore binária fazem com que existam diferenças sensíveis entre os níveis de suas folhas, gerando grandes diferenças de performance no acesso aos nodos da árvore.

Uma árvore é dita *perfeitamente balanceada* se para cada nodo da árvore o número de nodos da sua subárvore esquerda e direita diferem no máximo em 1.



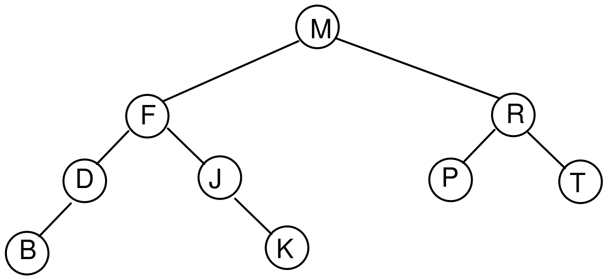
São exemplos de árvores balanceadas as árvores AVL (Adelson-Velskii e Landis, 1962).

ÁRVORE BINÁRIA DE BUSCA

Árvores binárias são freqüentemente usadas para representar um conjunto de dados cujos elementos são recuperados a partir de uma *chave* única. Caso a árvore seja organizada de tal maneira que para cada nodo v_i todas as chaves na subárvore esquerda de v_i são menores que v_i e as chaves na subárvore direita são maiores que a chave v_i, a árvore é chamada *árvore de busca*.

Em uma *árvore de busca* é possível localizar uma chave arbitrária iniciando pela raiz da árvore e prosseguindo ao longo de um caminho de pesquisa, alternando para o nodo da subárvore esquerda ou direita, dependendo de uma decisão baseada na avaliação da chave do nodo inspecionado.

A árvore abaixo representa uma árvore binária de busca gerada a partir da seqüência de chaves M, R, P, F, J, D, B,T, K.



O caminhamento de uma árvore binária de busca em ordem *inordem* fornece a relação de seus elementos em ordem ascendente.

- inserção em árvore binária de busca

- retirada em árvore binária de busca