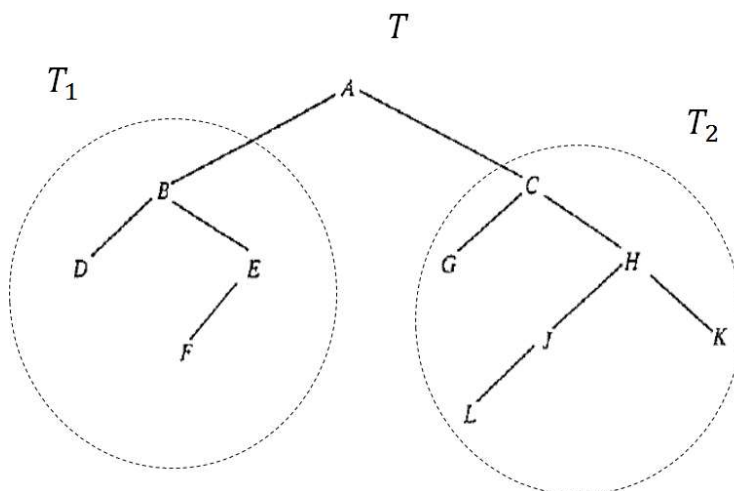


## Árvores binárias

Uma **árvore binária**  $T$  é definida como um conjunto finito de elementos, denominados por nós, tais que:

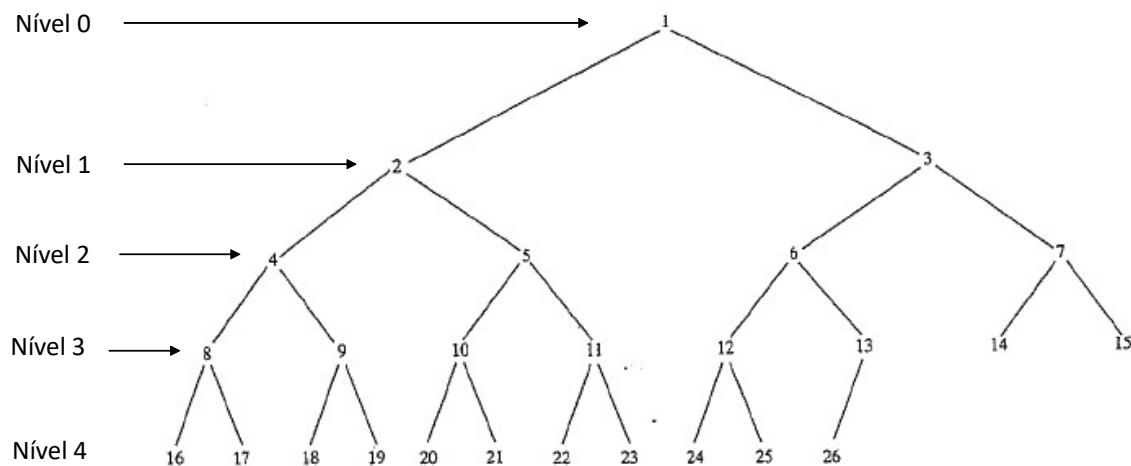
1.  $T$  é o conjunto vazio, ou
2.  $T$  contém um nó  $R$ , diferenciado dos restantes, a que chamamos raiz, e os outros nós de  $T$  formam um par ordenado de árvores binárias disjuntas,  $T_1$  e  $T_2$ .

Exemplo: Na figura seguinte podemos ver a árvore binária  $T$  com raiz  $A$  e as suas duas subárvores  $T_1$  e  $T_2$ . Podemos observar que as duas subárvores esquerda e direita do nó  $L$  são vazias e que o filho à esquerda do nó  $H$  é o  $J$  e o filho à direita do nó  $H$  é o  $K$ .



Uma **árvore binária**  $T$  é dita **completa** se todos os seus níveis, exeto possivelmente o último, têm o maior número possível de nós e se todos os nós do último nível aparecem o mais à esquerda possível.

Exemplo: na figura seguinte pode visualizar um exemplo de uma árvore binária completa.



**Uma árvore binária de busca  $T$** , é uma árvore binária onde cada nó  $N$  da árvore tem a seguinte propriedade: o valor de  $N$  é maior do que qualquer valor existente na subárvore esquerda de  $N$  e é menor ou igual do que qualquer valor existente na subárvore direita de  $N$ .

As árvores binárias de busca são uma das estruturas mais importantes na ciência da computação pois permitem localizar, inserir ou eliminar um elemento numa determinada lista com alguma facilidade.

### Representação sequencial de uma árvore binária

A representação sequencial de uma árvore binária  $T$  utiliza um único vetor linha, a que chamamos *ARVORE*:

1. a raiz  $R$  de  $T$  é armazenada em  $ARVORE(1)$ ;
2. se um nó  $N$  ocupa a posição  $k$  de  $ARVORE$  (falamos do elemento  $ARVORE(k)$ ), então o seu filho à esquerda é inserido na posição  $2k$ ,  $ARVORE(2k)$ , enquanto que o seu filho à direita ocupa a posição  $2k + 1$ ,  $ARVORE(2k + 1)$ .
3. caso o nó  $N$  não tenha um filho à esquerda ou um filho à direita, a respetiva posição em  $ARVORE$  fica vazia (coloca-se um 0 nesta posição para o simbolizar).

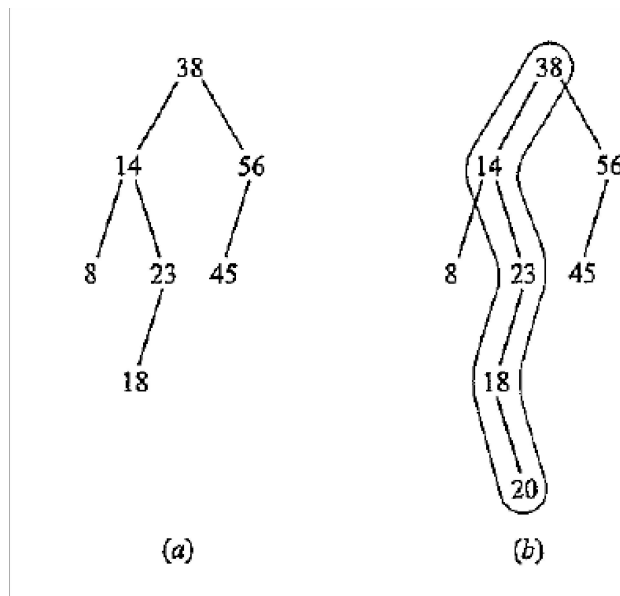
### Algoritmo para localização e inserção de um elemento

O algoritmo proposto serve para localizar um elemento, designado por *ITEM*, numa árvore binária de busca  $T$ ; caso este não seja encontrado, esse elemento é inserido na árvore.

1. Comparamos o *ITEM* com a raiz  $N$  da árvore  $T$ ;
  - 1.1. se  $ITEM < N$  então seguimos para o filho à esquerda de  $N$ ;
  - 1.2. se  $ITEM > N$  então seguimos para o filho à direita de  $N$ ;
2. Tomando  $N$  como o novo nó visitado, repetimos o ponto 1. até que uma das seguintes situações aconteça:
  - 2.1. encontrou-se um nó  $N$  tal que  $N = ITEM$ . Neste caso a busca foi bem sucedida e o algoritmo terminou;
  - 2.1 encontrou-se uma subárvore vazia. Como *ITEM* não foi localizado, este deverá ser inserido no lugar da subárvore vazia.

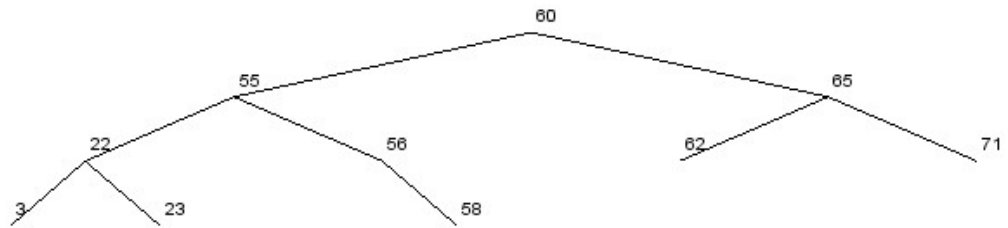
Exemplo:

Na figura seguinte, em (b) é possível visualizar o algoritmo de localização e inserção do  $ITEM = 20$  na árvore binária  $T$  representada em (a):



## Exercícios propostos

1. Considere a seguinte árvore binária:



- (a) Construa a representação sequencial da árvore binária e introduza-a no Scilab.
  - (b) Construa um algoritmo que permita verificar se a árvore dada é ou não uma árvore binária de busca, usando a função `em_ordem`, que percorre os nós da árvore considerando o percurso em-ordem.
2. Considere as listas de valores que se encontram no ficheiro "arvores.xls".
- (a) Construa uma função com o nome `insert_tree(Arvore,item)` que permita localizar um determinado `item` na árvore binária de busca `Arvore`; adicionalmente, caso o `item` não seja localizado, o algoritmo deverá inseri-lo na árvore, mantendo a estrutura de árvore binária de busca. Use esta função para inserir o `ITEM = 45` na árvore binária de busca que se encontra na folha designada por "lista 1" do ficheiro referido inicialmente.
  - (b) Construa um algoritmo que permita inserir, sequencialmente, numa árvore binária de busca vazia, os valores da tabela que se encontra na folha "lista2".