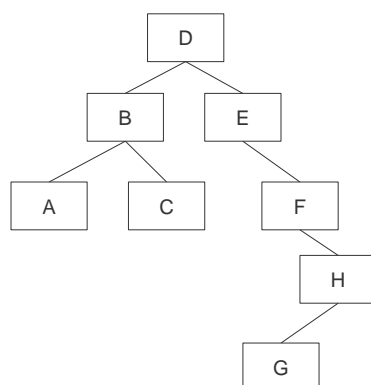


<p style="text-align: center;">Estruturas de Dados Ficha Laboratorial Nº 7 Engenharia Informática – Instituto Superior de Engenharia de Coimbra</p>
--

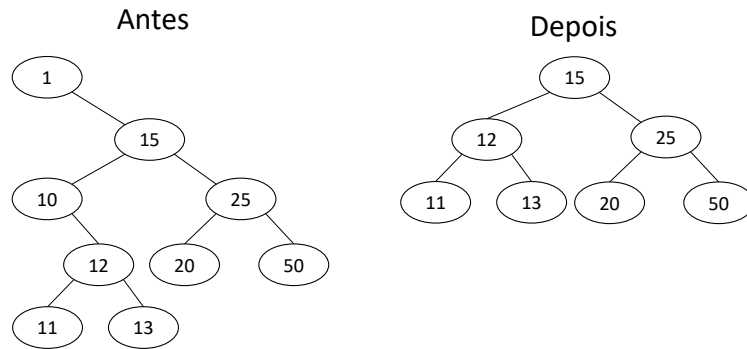
1 – Construa uma classe que implementa uma árvore binária de pesquisa, de acordo com a seguinte informação:

- a) A classe deve ser genérica. A comparação de elementos deverá ser feita através da comparação directa dos objectos inseridos, ou então através do uso de um comparador adequado fornecido aquando da construção da árvore.
- b) Deve disponibilizar os seguintes métodos:
 - a. `insere` : permite inserir um novo valor na árvore (gere uma excepção caso o elemento já se encontre na árvore)
 - b. `contém` : indica se um valor está ou não na árvore
 - c. `profundidadeDe`: indica a que profundidade se encontra um valor (a profundidade da raiz é 1), ou 0 se o elemento não existe.
 - d. `imprimeOrdem`: imprime os valores da árvore por ordem.
 - e. `Profundidade`: devolve a profundidade da árvore (a profundidade de uma árvore vazia é 0).
 - f. `Tamanho`: devolve o número de valores armazenados.
 - g. `ImprimeNíveis`: imprime o conteúdo da árvore nível a nível. (sugestão: use uma Fila auxiliar).
- c) Teste todos os métodos anteriores de forma adequada.

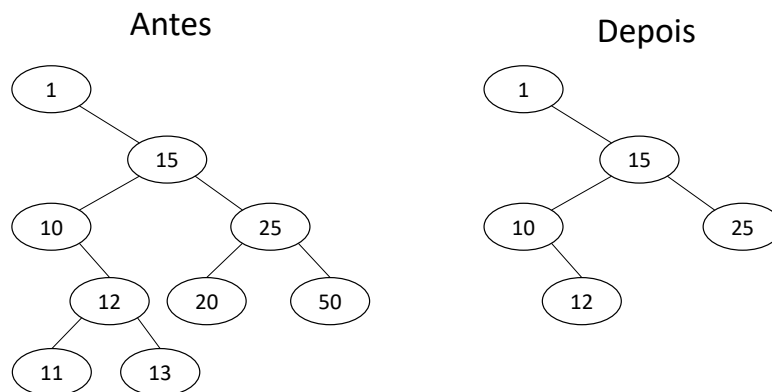
2- Construa uma árvore com o seguinte conteúdo (os dados são Strings):



- 3 – Utilize a função `imprimeNíveis` para verificar se a árvore anterior foi correctamente construída.
- 4 – Crie um comparador que compara strings de acordo com o seu comprimento. Crie uma árvore que usa esse comparador e teste o resultado.
- 5- Modifique a classe de forma a utilizar uma sentinela para nodos nulos.
- 6- Crie várias árvores aleatórias com 1000000 elementos e verifique a sua profundidade. Qual é o desvio em relação ao valor teórico “ideal”? (sugestão: inicialize uma lista ou array com os números inteiros entre 1 e 10000000 e baralhe-a, introduzindo depois esses valores na árvore)
- 7- Acrescente um método que remove um elemento da árvore.
- 8- Crie um método recursivo privado (e respetivo método público) que remove todos os nodos com um (e apenas um) descendente (este método não utiliza o método da pergunta anterior)



9 - Crie um método recursivo privado (e respetivo método público) que remove todas as folhas da árvore. (este método não usa o método definido na pergunta 7)



10 - Crie um método recursivo que recebe um valor X e devolve o sucessor de X na árvore (menor valor da árvore que é maior do que X), ou *null* se este não existir.

11 - Crie um método recursivo privado (e respetivo método para acesso público) que verifica se esta árvore é igual a outra recebida por parâmetro (duas árvores são iguais se tiverem o mesmo conteúdo (de acordo com o comparador usado) e a mesma estrutura).

12- Modifique a classe árvore de forma adequada para que passe a ser iterável. O iterador deve percorrer os elementos em ordem.