

Para efeitos da nota atribuída à resolução de exercícios ao longo do semestre - **Submeter até 23:59 de 12 de Junho** (o problema continuará depois disponível para submissão, mas sem contar para a nota)  
[para perceber o contexto do problema deve [ler o guião da aula #13](#)]

## [ED210] Será uma árvore de pesquisa?

Neste problema deverá apenas submeter uma classe **BSTree<T>** (e não um programa completo).

### Código Base

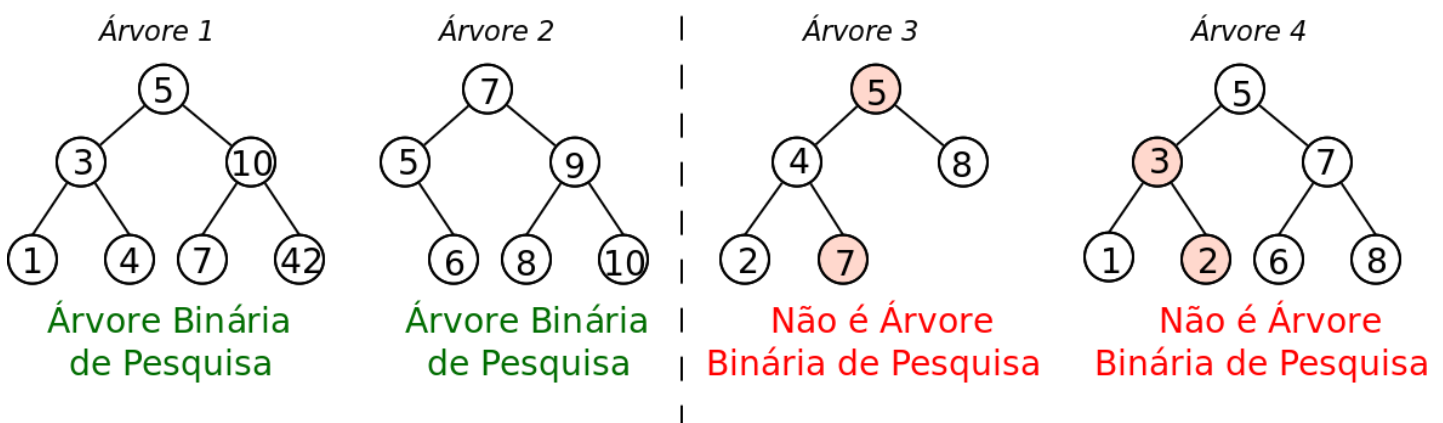
Use como base a classe **BSTree<T>** ([ver código](#) | download de [BSTNode.Java](#) e [BSTree.Java](#)), que representa uma árvore binária de pesquisa, tal como dada nas aulas).

### O problema

Acrescente à classe dada um método **public boolean valid()** que verifique se a árvore binária é realmente uma árvore de pesquisa, ou seja:

- Todas os valores (se existir algum) da subárvore esquerda são menores do que o valor da raiz
- Todas os valores (se existir algum) da subárvore direita são maiores do que o valor da raiz
- As subárvores esquerda e direita são também árvores binárias de pesquisa

A figura seguinte ilustra alguns exemplos de árvores binárias, indicando se são ou não árvores binárias de pesquisa:



- A árvore 3 não é de pesquisa pois 7 é maior que 5 mas está na sua subárvore esquerda.
- A árvore 4 não é de pesquisa pois 2 é menor que 3 mas está na sua subárvore direita.

Pode assumir que já foi lida uma árvore binária não vazia contendo somente elementos diferentes (não existem valores iguais) e agora só tem de verificar se é ou não de pesquisa.

### Submissão

Deverá submeter apenas a classe **BSTree<T>**, acrescentando o método **valid** como pedido (**e sem apagar ou modificar nenhum dos outros métodos dados como base**). Pode assumir que terá acesso no Mooshak à classe **BSTNode<T>** (não a pode mudar) e se precisar pode criar outros métodos auxiliares. O Mooshak irá criar várias instâncias da sua classe e irá fazer uma série de testes ao método por si implementado.

### Exemplos de Input/Output

Os exemplos correspondem às árvores da figura do enunciado.

Árvore <i>t</i> (em preorder com N a ser uma subárvore nula)	<i>t.valid()</i>
5 3 1 N N 4 N N 10 7 N N 42 N N	true

7 5 N 6 N N 9 8 N N 10 N N	true
5 4 2 N N 7 N N 8 N N	false
5 3 1 N N 2 N N 7 6 N N 8 N N	false

---

Estruturas de Dados (CC1007)  
DCC/FCUP - Faculdade de Ciências da Universidade do Porto

---