

[ED209] Intervalo de valores

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

Código Base

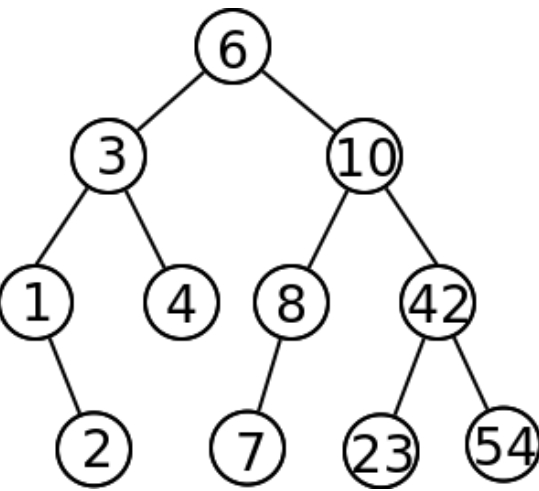
Use como base a classe **BSTree<T>** ([ver código](#) | download de [BSTNode.Java](#) e [BSTree.Java](#)), que representa uma árvore binária de pesquisa, tal como dada nas aulas).

O problema

Acrescente à classe dada um método **public int countBetween(T a, T b)** que devolve a **quantidade de elementos que são $\geq a$ e $\leq b$** , ou seja, que estão no intervalo $[a,b]$.

É garantido que a árvore binária é de pesquisa e que não é vazia (tem pelo menos um valor armazenado. É também garantido que $a \leq b$

A figura seguinte ilustra uma árvore binária de pesquisa.



Se a árvore for chamada de t , então alguns exemplos de chamadas seriam:

- $t.countBetween(5,44) = 6$ (valores 6, 7, 8, 10, 23 e 42)
- $t.countBetween(7,10) = 3$ (valores 7, 8 e 10)
- $t.countBetween(0,90) = 11$ (todos os valores da árvore)
- $t.countBetween(60,80) = 0$ (nenhum valor da árvore)

Submissão

Deverá submeter apenas a classe **BSTree<T>**, acrescentando o método **countBetween** 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 à árvore da figura do enunciado.

Chamada	Valor de retorno
t.countBetween(5,44)	6
t.countBetween(7,10)	3
t.countBetween(0,90)	11
t.countBetween(60,80)	0
t.countBetween(8,54)	5
t.countBetween(2,22)	7
t.countBetween(2,2)	1