

- 1) Considerando a declaração do tipo algébrico de dados `Tree` que representa uma árvore binária, considerando também que a função de inserção mantém a propriedade de uma árvore binária de pesquisa, declare uma função para retirar um elemento de uma árvore mantendo a propriedade da árvore binária de pesquisa. O elemento a ser retirado deve ser passado como parâmetro.

```
data Tree a = Node a (Tree a) (Tree a) | Leaf deriving Show
```

```
ins e Leaf = Node e Leaf Leaf
ins e n@(Node x l r)
  | e == x = n
  | e < x  = Node x (ins e l) r
  | e > x  = Node x l (ins e r)
```

- 2) Considerando a declaração do tipo de dado algébrico `TreeRB` que representa uma árvore vermelha e preta, declare uma função de inserção que respeite as propriedades deste tipo de árvore:

- I. Todo nó é vermelho ou preto.
- II. As folhas são pretas.
- III. Se um nó é vermelho seus filhos devem ser pretos.
- IV. Todos os caminhos partindo da raiz até as folhas contém o mesmo número de nós pretos.
- V. Os novos nós devem ser criados como vermelhos.

```
data RB = R | B deriving Show
data RBTree a = LeafRB | Node a RB (RBTree a) (RBTree a) deriving Show
```