



Faculdade de Computação
Programação Funcional (BCC/BSI)
5a. Lista de Exercícios: Listas, Árvore Binária, Ordenação

1. Defina uma função que, dada uma lista numérica, retorna uma tupla-2, tal que contenha o maior valor da lista, bem como sua posição relativa.

Exemplo:

```
> maior [13,8,15,12,26,4]
(26,5)
```

2. Seja a definição abaixo para uma estrutura de dados em árvore binária de busca.

```
data ArvBinInt = Nil | NoInt Int ArvBinInt ArvBinInt deriving Show

arvDados :: ArvBinInt
arvDados = NoInt 7 (NoInt 5 (NoInt 2 Nil Nil) (NoInt 6 Nil Nil))
               (NoInt 13 (NoInt 9 Nil Nil) (NoInt 20 Nil (NoInt 23 Nil Nil)))
```

a) Desenhe a árvore de dados correspondente à arvDados. Em seguida faça funções para:

- a1) verificar se uma árvore é vazia
- a2) retornar a sub-árvore à esquerda de um nó (raiz da árvore)
- a3) inserir ordenadamente um novo elemento (repetições devem ser evitadas)

b) Seja a função abaixo de retorno e remoção do maior elemento. Verifique se está correta e caso necessário faça as devidas correções. Explique a estratégia a ser utilizada na remoção de um elemento qualquer em árvore binária de busca.

```
deletemax :: ArvBinInt -> (Int, ArvBinInt)
deletemax (NoInt y t1 Nil) = (y, t1)
deletemax (NoInt y t1 t2) = (z, NoInt y t1 tz)
    where (z, tz) = deletemax t2
```

3) Seja uma agência bancária que deseja vender títulos de capitalização para seus correntistas. O título é adquirido em apenas uma parcela. Cada correntista pode comprar um título (apenas um). Cada título de capitalização deve ser definido por uma tupla com os seguintes dados:

- Numero da Conta
- Data da compra (tupla com dia, mês e ano)
- Valor escolhido (300, 500, 1000 e 1500)

Assim, podemos ter os exemplos abaixo como tuplas para títulos de capitalização:

```
("689232", (25,10,2011), 1000)
("236333", (27,10,2011), 300)
("546544", (08,11,2011), 1500)
("975457", (16,11,2011), 1000)
```

- A) Crie uma estrutura de **árvore de busca binária** para armazenar as compras de títulos de capitalização dos correntistas.
- B) Defina uma função **inserir** para guardar cada título na árvore, usando o número da conta como elemento identificador. A função **inserir** deve garantir que não seja registrado mais de um título por correntista.
- C) Ao final do prazo para a venda dos títulos devem-se começar os sorteios, que acontecem semanalmente. O valor a pagar ao correntista sorteado é 1000 vezes o valor escolhido. Para isso:

C1) Faça uma função que retira os elementos (apenas os números da conta) da árvore de busca binária, e os coloca numa lista, usando a estratégia da pós-ordem.

C2) Usando um número qualquer como entrada, faça uma função **sorteio**, que realiza um percurso circular na lista e devolve o número da conta sorteada. No percurso circular, deve-se, ao chegar no final da lista, começar novamente do início da mesma até que o número de entrada seja igual a um.

Ex: > sorteio 10 ["689232", "236333", "546544", "975457"]
 "236333"

C3) Para cada conta sorteada, removê-la da árvore binária para que um novo sorteio possa ser realizado.