

Paradigmas da Programação I

ESI

Ano Lectivo de 2005/2006 (Época Especial)

Questão 1 Relembre o tipo `BTree` a definido para representar árvores binárias.

```
data BTree a = V | N a (BTree a) (BTree a)
```

1. Defina uma função `bf :: BTree a -> [a]` de travessia de uma árvore por níveis. Para a árvore

```
a = N 1 (N 2 (N 3 V V) (N 4 V V)) (N 5 (N 6 V V) (N 7 V V))
```

devemos ter que `bf a = [1,2,5,3,4,6,7]`

2. Relembre ainda a função `inorder :: Btree a -> [a]` assim definida.

```
inorder V = []
inorder (N x e d) = (inorder e) ++ [x] ++ (inorder d)
```

Defina uma função `etiqueta :: BTree a -> BTree (a,Int)` que satisfaça a seguinte propriedade:

```
inorder (etiqueta a) = zip (inorder a) [1..]
```

Questão 2 Considere o seguinte tipo para representar expressões numéricas.

```
data Exp = C Int | V String | Op OBin Exp Exp
```

```
data OBin = Mais | Menos | Vezes | Div
```

Neste tipo a expressão $(x + 1) - (1 - x)$ pode ser representada por

```
e = Op Menos (Op Mais (V "x") (C 1)) (Op Menos (C 1) (V x))
```

1. Defina uma função `freeV :: Exp -> [String]` que dá a lista (sem repetições) das variáveis que ocorrem numa expressão.
2. Defina uma função `substitui :: Exp -> [(String,Exp)] -> Exp` que aplica uma dada substituição a uma expressão. Por exemplo, se a expressão acima for designada por `e`, o resultado de `substitui e [("x",e)]` é a expressão

```
((((x + 1) - (1 - x)) + 1) - (((x + 1) - (1 - x)) - 1))
```

Questão 3 A função `nub :: (Eq a) => [a] -> [a]` pode ser definida por:

```
nub [] = []
nub (h:t) = h:(nub (filter (/= h) t))
```

1. Podemos generalizar esta função passando-lhe como argumento um predicado binário. Temos então a seguinte definição.

```
nubBy :: (a -> a -> Bool) -> [a] -> [a]
nubBy _ [] = []
nubBy p (h:t) = h:(nubBy (filter (not . (p h) ) t))
```

Defina a função `nub` usando esta generalização.

2. Usando ainda a função `nubBy`, defina uma função `nub3` que, dada uma lista de ternos, calcula essa lista sem repetições na segunda componente. Por exemplo

```
nub3 [(1,2,3), (1,4,3), (2,2,3), (2,4,4)] = [(1,2,3), (1,4,3)]
```

Qual o tipo de `nub3`?