

Considerem aquesta definició de tipus algebraic genèric per a arbres binaris amb talla (nombre de nodes del subarbre a cada node):

```
data STree a = Nil | Node Int a (STree a) (STree a) deriving Show
```

1. Escriviu una funció `isOk :: STree a → Bool` que indiqui si les talles dels nodes d'un arbre amb talles són correctes.
2. Escriviu una funció `nthElement :: STree a → Int → Maybe a` que retorni l'*n*-èsim element en inordre (començant per 1) d'un arbre amb talla correcta, o **Nothing** si no existeix. El cost ha de ser $O(h)$ on *h* és l'alçada de l'arbre.
3. Escriviu una funció `mapToElements :: (a → b) → STree a → [Int] → [Maybe b]` que aplica (potser) una funció *a* a una llista d'elements d'un arbre amb talla correcta (identificats per la seva posició en inordre).
4. Feu que `STree` sigui un functor.

Fixeu-vos en els exemples.

Exemple d'entrada

```
let div10 = flip div 10
let t1 = Node 3 99 (Node 1 88 Nil Nil) (Node 1 22 Nil Nil)
let t2 = Node 2 77 (Node 1 33 Nil Nil) Nil
let t3 = Node 6 44 t1 t2
let t4 = Node 7 55 t1 t2
isOk t3
isOk t4
nthElement t3 1
nthElement t3 9
nthElement t3 3
nthElement t3 8
mapToElements div10 t3 [1,9,3,8]
div10 <$> t3
```

Exemple de sortida

```
True
False
Just 88
Nothing
Just 22
Nothing
[Just 8,Nothing,Just 2,Nothing]
Node 6 4 (Node 3 9 (Node 1 8 Nil Nil) (Node 1 2 Nil Nil)) (Node 2 7 (Node 1 3 Nil Nil) Nil)
```

Informació del problema

Autor : Jordi Petit, Gerard Escudero

Generació : 2024-05-02 20:10:43

© Jutge.org, 2006–2024.

<https://jutge.org>