99 questions/Solutions/62

From HaskellWiki

< 99 questions | Solutions

Collect the internal nodes of a binary tree in a list

An internal node of a binary tree has either one or two non-empty successors. Write a predicate internals/2 to collect them in a list.

```
internals :: Tree a -> [a]
internals Empty                  = []
internals (Branch a Empty Empty) = []
internals (Branch a left right ) = a : internals left ++ internals right
```

Alternative solution only using cons:

```
internals t = internals' t []
    where internals'  Empty                 xs = xs
          internals' (Branch x Empty Empty) xs = xs
          internals' (Branch x l r)         xs = (x :) $ internals' l $ internals' r xs
```

Retrieved from "https://wiki.haskell.org/index.php?title=99_questions/Solutions/62&oldid=47703"

- This page was last modified on 1 August 2012, at 13:06.
-