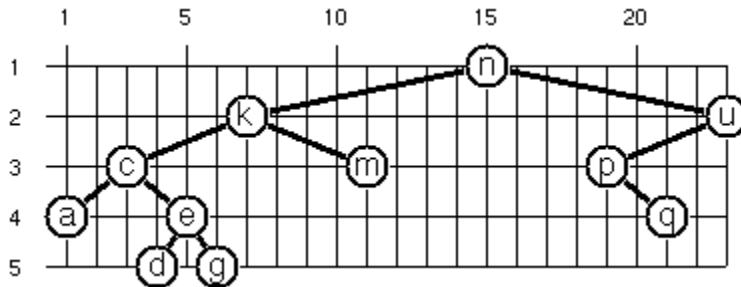


99 questions/Solutions/65

From HaskellWiki

< 99 questions | Solutions

An alternative layout method is depicted in the illustration below:



Find out the rules and write the corresponding function. Hint: On a given level, the horizontal distance between neighboring nodes is constant.

Use the same conventions as in problem P64 and test your function in an appropriate way.

```
layout :: Tree a -> Tree (a, Pos)
layout t = layoutAux x1 1 sep1 t
  where d = depth t
        ld = leftdepth t
        x1 = 2^(d-1) - 2^(d-ld) + 1
        sep1 = 2^(d-2)
        layoutAux x y sep Empty = Empty
        layoutAux x y sep (Branch a l r) =
          Branch (a, (x,y))
            (layoutAux (x-sep) (y+1) (sep `div` 2) l)
            (layoutAux (x+sep) (y+1) (sep `div` 2) r)
```

```
depth :: Tree a -> Int
depth Empty = 0
depth (Branch a l r) = max (depth l) (depth r) + 1
```

```
leftdepth :: Tree a -> Int
leftdepth Empty = 0
leftdepth (Branch a l r) = leftdepth l + 1
```

The auxiliary function is passed the x- and y-coordinates for the root of the subtree, the horizontal separation between the root and its child nodes, and the subtree itself. It returns the subtree annotated with positions.

Another solution is presented below. The root is placed at (0, 0). (I assume this

solution uses more memory, since we pass the list of children into the recursion for each subtree. Can somebody verify this?)

```
tree65 t =
  helper [t] 0 0 t
  where
    helper ss x y Empty =
      Empty
    helper ss x y (Branch _ t0 t1) =
      let
        r = foldr1 max $ map radius ss
        ss' = concatMap children ss
      in
        Branch (x, y) (helper ss' (x - r) (y + 1) t0) (helper ss' (x + r) (y + 1) t1)
```

```
children (Branch _ t0 t1) =
  [t0, t1]
children Empty =
  []
```

```
radius Empty =
  1
radius (Branch _ t0 t1) =
  let
    w0 = 2 * (radius t0) - 1
    w1 = 2 * (radius t1) - 1
  in
    max w0 w1 + 1
```

Retrieved from "https://wiki.haskell.org/index.php?title=99_questions/Solutions/65&oldid=40821"

- This page was last modified on 4 July 2011, at 11:51.
- Recent content is available under a simple permissive license.