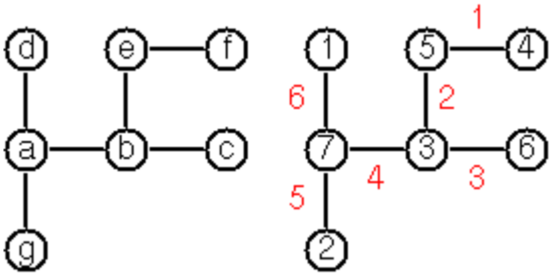
99 questions/Solutions/92

From HaskellWiki

< 99 questions | Solutions

(***) Von Koch's conjecture

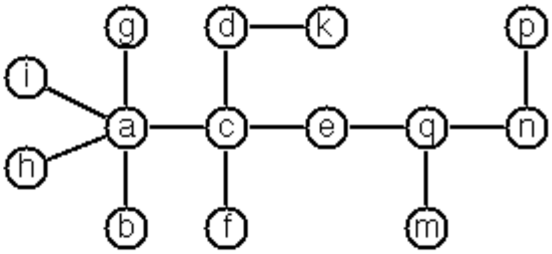
Several years ago I met a mathematician who was intrigued by a problem for which he didn't know a solution. His name was Von Koch, and I don't know whether the problem has been solved since.



Anyway the puzzle goes like this: Given a tree with N nodes (and hence N-1 edges). Find a way to enumerate the nodes from 1 to N and, accordingly, the edges from 1 to N-1 in such a way, that for each edge K the difference of its node numbers equals to K. The conjecture is that this is always possible.

For small trees the problem is easy to solve by hand. However, for larger trees, and 14 is already very large, it is extremely difficult to find a solution. And remember, we don't know for sure whether there is always a solution!

Write a predicate that calculates a numbering scheme for a given tree. What is the solution for the larger tree pictured below?



Solution:

```
vonKoch edges = do
    let n = length edges + 1
    nodes <- permutations [1..n]
    let nodeArray = listArray (1,n) nodes
```

```haskell
    let dists = sort $ map (\(x,y) -> abs (nodeArray ! x - nodeArray ! y)) edges
    guard $ and $ zipWith (/=) dists (tail dists)
    return nodes
```

This is a simple brute-force solver. This function will permute all assignments of the different node numbers and will then verify that all of the edge differences are different. This code uses the List Monad.

Retrieved from "https://wiki.haskell.org/index.php?title=99_questions/Solutions/92&oldid=36209"