```
sig Node {
 children : set Node
}
sig Leaf extends Node {}
one sig Root in Node {}

sig Red, Black in Node {}

pred Invs {
 // Specify the properties that characterize
    // red-black binary trees inside this predicate.

 // The number of points you will get is proportional to the number of correct properties.
 // To check how many points you have so far you can use the different commands.
    // The maximum is 5 points.

 // Be careful to not overspecify!
 // If some of your properties are not valid in a red-black tree you get 0 points,
    // even if you have some correct properties.
 // To check if you are not overspecifying you can use command NoOverspecification.
 // If you are overspecifying this command will return a tree that should be possible
 // but that you spec is not accepting.

   //Nodes-----------------------------------------------------------------
   //Every node is either red or black.
   all n: Node | n in Red iff n not in Black

   //A node can't have itself as a child
   all n : Node | n not in n.^children
   //----------------------------------------------------------------------

   //Root------------------------------------------------------------------
   //The Root is always black.
   all r: Root | r in Black

   //The root doesn't have father
   all r: Root | no(children.r)
   //----------------------------------------------------------------------

   //Leafs-----------------------------------------------------------------
   //A leaf doesn't have children
   all l : Leaf | no(l.children)

   //A leaf is black.
   all l : Leaf | l in Black
   //----------------------------------------------------------------------

   //All nodes but [...]---------------------------------------------------
   //All nodes except leafs have two children
   all n: Node-Leaf | #(n.children) = 2

   //All nodes except the root have one father
   all n: Node-Root | #(children.n) = 1

   //All nodes except root & leafs (interior nodes) have 2 children
   all i: Node-Root-Leaf | #(i.children) = 2
   //----------------------------------------------------------------------

   //Other invariants------------------------------------------------------
   //Both children of red nodes are black
   all c: Node-Black | c.children in Black

   //All leafs must have the same number of black nodes from root (including) to themselves
   all l1,l2: Leaf| #((^children).l1 & Black) = #((^children).l2 & Black)
   //----------------------------------------------------------------------
}
```