

```

-- 1
{--

LET False = \ a b. b
LET (?:) = \ c t f . c (\ x . t) f
LET GTE   = \ a b . iszero (sub b a) ? True : False

--}

-- 2

data Tree k d
  = Nil
  | Nd k d (Tree k d) (Tree k d)

find :: Ord k => k -> Tree k d -> Maybe d
find _ Nil = Nothing
find k (Nd k' d' l r)
  | k==k' = Just d'
  | k<k' = find k l
  | True = find k r

-- 3
{--

sum [] = 0                -- d1
sum (x:xs) = x + sum xs   -- d2

[] ++ ys = ys             -- d3
(x:xs) ++ ys = x:(xs ++ys) -- d4

1) xs==[]
L = sum ([]++ys) =|3 sum ys
P = sum [] + sum ys =|1 0 + sum ys =|aritm0+ sum ys
L = P

2)
I.H. sum (as++ys) = sum as + sum ys
xs = (a:as)

L = sum ((a:as)++ys) =|4 sum (a:(as++ys)) =|2 a + sum (as++ys)
P = sum (a:as) + sum ys =|2 a + sum as + sum ys =|I.H. a + sum (as++ys)
L = P

Q.E.D.
--}

-- EOF

```