```
let rec insert n lst =
  match lst with
  | [] -> [n]
  | x :: xs -> if (n < x) then n:: lst else x::(insert n xs)

let rec isort lst =
  match lst with
  | [] -> []
  | x :: xs -> insert x (isort xs)
```

We assume that insert is correct, meaning that it inserts the term n at the right place in the list.

Base case: empty list []

isort []

-> [] by program isort

induction hypothesis: assume this holds for a list of length up to n

induction step: see if this holds for n+1 or x::t

isort x::t::[]

-> insert x (isort t::[]) by program isort

-> insert x (insert t (isort [])) by program isort

-> insert x (insert t []) by program isort

-> insert x [t] by program insert