189 (fixed point) Let L be a nonempty sorted list of different integers. Write a program to find a fixed-point of L, that is an index i such that $Li = i$, or to report that no such index exists. Execution time should be at most $log\ (\#L)$.

§ Let L be a constant, and let i and j be a natural variables. Let t be an extended natural time variable. If a fixed-point exists, it will be indicated by $Li'=i'$. If none exists, that will be indicated by $Li'\neq i'$.

$$(\exists k\colon 0,..\#L\cdot Lk=k) = (Li'=i') \;\Leftarrow\; i:= 0.\; j:= \#L.\; i<j \;\Rightarrow\; (\exists k\colon i,..j\cdot Lk=k) = (Li'=i')$$

$i<j \;\Rightarrow\; (\exists k\colon i,..j\cdot Lk=k) = (Li'=i') \;\Leftarrow$
  if $j–i=1$ **then** ok
  else $m:= div\ (i+j)\ 2.$
    if $Lm\le m$ **then** $i:= m$ **else** $j:= m$ **fi**.
    $i<j \;\Rightarrow\; (\exists k\colon i,..j\cdot Lk=k) = (Li'=i')$ **fi**

The timing:

$$t' \le t + ceil\ (log\ (\#L)) \;\Leftarrow\; i:= 0.\; j:= \#L.\; i<j \Rightarrow t' \le t + ceil\ (log\ (j–i))$$

$i<j \Rightarrow t' \le t + ceil\ (log\ (j–i)) \;\Leftarrow$
  if $j–i=1$ **then** ok
  else $m:= div\ (i+j)\ 2.$
    if $Lm\le m$ **then** $i:= m$ **else** $j:= m$ **fi**.
    $t:= t+1.\; i<j \Rightarrow t' \le t + ceil\ (log\ (j–i))$ **fi**

The first refinement is proven by two uses of the Substitution Law. The last refinement is proven in three cases. First case:

$(i<j \;\Rightarrow\; (\exists k\colon i,..j\cdot Lk=k) = (Li'=i') \;\Leftarrow\; j–i=1 \wedge ok)$ — expand ok
$= (i<j \;\Rightarrow\; (\exists k\colon i,..j\cdot Lk=k) = (Li'=i') \;\Leftarrow\; j–i=1 \wedge i'=i \wedge j'=j)$ — context
$= (i<i+1 \;\Rightarrow\; (\exists k\colon i,..i+1\cdot Lk=k) = (Li=i) \;\Leftarrow\; j–i=1 \wedge i'=i \wedge j'=j)$ — simplify
$= (\top \;\Rightarrow\; (Li=i) = (Li=i) \;\Leftarrow\; j–i=1 \wedge i'=i \wedge j'=j)$ — reflexive and identity
$= \top$

Middle case:

$(\quad i<j \;\Rightarrow\; (\exists k\colon i,..j\cdot Lk=k) = (Li'=i')$
$\Leftarrow\; j–i\neq 1 \wedge (m:= div\ (i+j)\ 2.\; Lm\le m \wedge (i:= m.\; i<j \;\Rightarrow\; (\exists k\colon i,..j\cdot Lk=k) = (Li'=i'))))$
— portation

$= \quad j–i\ge 2 \;\wedge\; (m:= div\ (i+j)\ 2.\; Lm\le m \wedge (i:= m.\; i<j \;\Rightarrow\; (\exists k\colon i,..j\cdot Lk=k) = (Li'=i')))$
$\Rightarrow (\exists k\colon i,..j\cdot Lk=k) = (Li'=i')$ — Substitution Law twice

$= \quad j–i\ge 2 \;\wedge\; L(div\ (i+j)\ 2)\le(div\ (i+j)\ 2)$
$\wedge\; ((div\ (i+j)\ 2)<j \;\Rightarrow\; (\exists k\colon (div\ (i+j)\ 2),..j\cdot Lk=k) = (Li'=i'))$
$\Rightarrow (\exists k\colon i,..j\cdot Lk=k) = (Li'=i')$
— In the context $j–i\ge 2$, we have $(div\ (i+j)\ 2)<j$, so discharge

$= \quad j–i\ge 2 \;\wedge\; L(div\ (i+j)\ 2)\le(div\ (i+j)\ 2) \;\wedge\; (\exists k\colon (div\ (i+j)\ 2),..j\cdot Lk=k) = (Li'=i')$
$\Rightarrow (\exists k\colon i,..j\cdot Lk=k) = (Li'=i')$
— If $L(div\ (i+j)\ 2)=(div\ (i+j)\ 2)$, then $(\exists k\colon (div\ (i+j)\ 2),..j\cdot Lk=k)$ and $(\exists k\colon i,..j\cdot Lk=k)$ are both $\top$.
If $L(div\ (i+j)\ 2)<(div\ (i+j)\ 2)$, then $(\exists k\colon i,..(div\ (i+j)\ 2)\cdot Lk=k)$ is $\bot$ because L is strictly increasing.

$= \quad j–i\ge 2 \;\wedge\; L(div\ (i+j)\ 2)\le(div\ (i+j)\ 2) \;\wedge\; (\exists k\colon i,..j\cdot Lk=k) = (Li'=i')$
$\Rightarrow (\exists k\colon i,..j\cdot Lk=k) = (Li'=i')$ — specialization

$= \quad \top$

The last case is just like the middle case. The timing proof breaks into the same cases as the results proof.