Logarithmic execution time is a big clue that we should try a binary search. Look in the middle of the list; if there’s an equal pair there, great; if not, eliminate half the list and look in the other half. Or, modelling the solution after binary search, use the pair in the middle of the list to eliminate half the list without checking to see if it’s an equal pair.

Let the list be \( L \). Suppose, for some \( m: 2..\#L \), that \( L(m-1) > Lm \). Because \( L \) is convex, the first line below is a theorem, and therefore so is the last line.

\[
L(m-1) \leq (L(m-2)+Lm)/2 \quad \text{increase \( Lm \) to \( L(m-1) \)}
\]

\[
L(m-1) < (L(m-2)+L(m-1))/2
\]

\[
L(m-2) > L(m-1)
\]

By the same reasoning, all previous pairs are decreasing, and therefore unequal. Similarly if \( L(m-1) < Lm \) all following pairs are unequal. The question does not ask where a consecutive pair of equal items is, but only whether there is such a pair. So my specification is \( R \), where

\[
R \equiv p' = (\exists i: 1..\#L \cdot L(i-1) = Li)
\]

Introduce indexes \( h \) and \( j \) and define

\[
Q \equiv 0 < h < j \leq \#L \Rightarrow p' = (\exists i: h..j \cdot L(i-1) = Li)
\]

We now solve the problem.

\[
R \iff h:= 1. j:= \#L. Q
\]

\[
Q \iff \text{if } j-h = 1 \text{ then } p:= L(h-1) = Lh
\]

\[
\text{else } m:= \text{div}(h+j)/2.
\]

\[
\text{if } L(m-1) > Lm \text{ then } h:= m \text{ else } j:= m \cdot fi.
\]

\[
Q \cdot fi
\]

Here is the proof of the first refinement, starting with the right side.

\[
h:= 1. j:= \#L. Q \quad \text{substitution law twice}
\]

\[
0 < 1 < \#L \leq \#L \Rightarrow p' = (\exists i: 1..\#L \cdot L(i-1) = Li) \quad L \text{ is convex therefore } 1 < \#L
\]

\[
R
\]

For the second refinement, the main else becomes

\[
m:= \text{div}(h+j)/2.
\]

\[
\text{if } L(m-1) > Lm \text{ then } h:= m \text{ else } j:= m \cdot fi.
\]

\[
Q \iff \text{if } j-h = 1 \text{ then } p:= L(h-1) = Lh
\]

\[
\text{else if } L((\text{div}(h+j)/2) - 1) > L((\text{div}(h+j)/2) \text{ then } m:= \text{div}(h+j)/2. h:= m. Q \quad \text{substitution law}
\]

\[
\text{else } m:= \text{div}(h+j)/2. j:= m. \cdot fi.
\]

So the refinement

\[
Q \iff \text{if } j-h = 1 \text{ then } p:= L(h-1) = Lh
\]

\[
\text{else if } L((\text{div}(h+j)/2) - 1) > L((\text{div}(h+j)/2) \text{ then } m:= \text{div}(h+j)/2. h:= m. Q
\]

\[
\text{else } m:= \text{div}(h+j)/2. j:= m. \cdot fi \cdot fi
\]

can now be treated in three cases. First case:

\[
Q \iff j-h = 1 \land (p:= L(h-1) = Lh) \quad \text{replace } Q
\]

\[
(0 < h < j \leq \#L \Rightarrow p' = (\exists i: h..j \cdot L(i-1) = Li)) \iff j-h = 1 \land (p:= L(h-1) = Lh)
\]

\[
0 < h < j \leq \#L \land j-h = 1 \land p' = (\exists i: h..j \cdot L(i-1) = Li) \quad \text{assignment}
\]

\[
0 < h < j \leq \#L \land j-h = 1 \land p' = (\exists i: h..j \cdot L(i-1) = Li) \land m'=m \land h'=h \land j'=j
\]

\[
\Rightarrow p' = (\exists i: h..j \cdot L(i-1) = Li) \quad \text{generalization, context}
\]

\[
\iff \top
\]

Middle case:
The timing analysis is identical to binary search.

As promised, we prove $0 < h < j \leq \#L$.

0. $h < j \leq \#L$.

As promised, we prove $0 < h < j \leq \#L$.

0. $h < j \leq \#L$.

As promised, we prove $0 < h < j \leq \#L$.

0. $h < j \leq \#L$.

As promised, we prove $0 < h < j \leq \#L$.

0. $h < j \leq \#L$.

As promised, we prove $0 < h < j \leq \#L$.

0. $h < j \leq \#L$.