The problem is \( P \), defined as
\[
P = p' = \text{MAX} i: 0..\#L+1\cdot \text{MAX} j: i..\#L+1\cdot \prod L[j: i..j]
\]
using int variable \( p \) for the answer. We also use variable \( k: \text{nat} \) as a list index, and variables \( c, d, x: \text{int} \). Define condition
\[
J = p = (\text{MAX} i: 0..k+1\cdot \text{MAX} j: i..k+1\cdot \prod L[i..j])
\]
\[
\wedge c = (\text{MAX} i: 0..k+1\cdot \prod L[i..k])
\]
\[
\wedge d = (\text{MIN} i: 0..k+1\cdot \prod L[i..k])
\]
Here are the refinements.
\[
P \iff p := 1. \ c := 1. \ d := 1. \ k := 0. \ J \Rightarrow P
\]
\[
J \Rightarrow P \iff \text{if } k = \#L \text{ then } \text{ok}
\]
\[
\begin{aligned}
&\text{else if } Lk \geq 0 \text{ then } c := \text{max} (c \times Lk) 1. \ d := \text{min} (d \times Lk) 1
\end{aligned}
\]
\[
\begin{aligned}
&\text{else } x := c. \ c := \text{max} (d \times Lk) 1. \ d := \text{min} (x \times Lk) 1 \text{ fi.}
\end{aligned}
\]
\[
p := \text{max} p \ c. \ k := k + 1. \ J \Rightarrow P \text{ fi}
\]
and the timing is
\[
t' = t + \#L \leftarrow p := 1. \ c := 1. \ d := 1. \ k := 0. \ t' = t + \#L - k
\]
\[
t' = t + \#L - k \iff \text{if } k = \#L \text{ then ok}
\]
\[
\begin{aligned}
&\text{else if } Lk \geq 0 \text{ then } c := \text{max} (c \times Lk) 1. \ d := \text{min} (d \times Lk) 1
\end{aligned}
\]
\[
\begin{aligned}
&\text{else } x := c. \ c := \text{max} (d \times Lk) 1. \ d := \text{min} (x \times Lk) 1 \text{ fi.}
\end{aligned}
\]
\[
p := \text{max} p \ c. \ k := k + 1. \ t := t + 1. \ t' = t + \#L - k \text{ fi}
\]
Proof of the first refinement: after 4 substitutions, \( J \) simplifies to \( \top \).
The second refinement breaks into 3 cases. Each case begins with portation, so we are proving
\[
J \wedge \text{something} \Rightarrow P
\]
by starting with the antecedent. First case:
\[
J \wedge k = \#L \wedge \text{ok}
\]
\[
i \text{ncot } k = \#L \wedge p' = p \text{ the first conjunct of } J \text{ is } P \Rightarrow P
\]
Second case:
\[
J \wedge k + \#L \wedge Lk \geq 0
\]
\[
\wedge (c := \text{max} (c \times Lk) 1. \ d := \text{min} (d \times Lk) 1). \ p := \text{max} p \ c. \ k := k + 1. \ J \Rightarrow P
\]
Make 4 substitutions. Note that \( P \) does not mention any of the 4 variables
(it mentions \( p' \) but not \( p \)).
I'm reversing \( J \Rightarrow P \) to \( P \Leftarrow J \) for typesetting reasons.
\[
= \quad J \wedge k + \#L \wedge Lk \geq 0
\]
\[
\wedge (P \Leftarrow (\text{max} p (\text{max} c \times Lk) 1) = (\text{MAX} i: 0..k+2\cdot \text{MAX} j: i..k+2\cdot \prod L[i..j])
\]
\[
\wedge \text{max} (c \times Lk) 1 = (\text{MAX} i: 0..k+2\cdot \prod L[i..k+1])
\]
\[
\wedge \text{min} (d \times Lk) 1 = (\text{MIN} i: 0..k+2\cdot \prod L[i..k+1])
\]
We need \( J \wedge k + \#L \wedge Lk \geq 0 \) to discharge the implication, so we need to show that it implies the antecedent of the implication. \( J \) says that \( p \) is the maximum product of all segments ending at or before \( k \), and that \( c \) is the maximum product of all segments ending at \( k \), and that \( d \) is the minimum product of all segments ending at \( k \) (remember that we write indexes between items). To find the maximum product of all segments ending at or before \( k + 1 \), we need only consider the new sequences, which are those ending at \( k + 1 \). One of them is the empty sequence whose product is 1. The others are all one-item extensions of sequences ending at \( k \). Since the new item \( Lk \) is nonnegative, the maximum product of these extensions is the maximum product \( c \) of these sequences ending at \( k \) times the new item \( Lk \). So the maximum product of all segments ending at or before \( k \) is the maximum of \( p \), \( c \times Lk \), and 1. That's what the first two conjuncts of the
antecedent say, so they are discharged. The last conjunct of the antecedent is also discharged by a similar argument. This hint is ridiculously long, informal, and inadequate.

The last case is much like the previous case, but slightly more complicated because $L_k$ is negative, and so multiplying by it switches maximums and minimums.

(b) the segment (sublist of consecutive items) whose product is maximum.