The problem is to find the maximum product of any segment (sublist of consecutive items).

Given a list of integers, possibly including negatives, write a program to find

$227$ (maximum product segment) Given a list of integers, possibly including negatives, write

a program to find

the maximum product of any segment (sublist of consecutive items).

The problem is

$P$ defined as

$P = p = {\prod}_{i:0..L+1} \cdot {\prod}_{j: i..L+1} \Pi \ L \ [i..j]$ 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

$J = p = {\prod}_{i:0..k+1} \cdot {\prod}_{j: i..k+1} \Pi \ L \ [i..j]$

and the timing is

$t' = t+\#L \iff p := 1 \cdot c:= 1 \cdot d:= 1 \cdot k:= 0 \cdot \Rightarrow J \Rightarrow P$

and the timing is

$t' = t+\#L \iff k=\#L \ then \ ok$

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:

$J \wedge k = \#L \wedge \text{ok}$

in context $k = \#L \wedge p' = p$ the first conjunct of $J$ is $P$

$\Rightarrow P$

Second case:

$J \wedge k+\#L \wedge L \ k \geq 0$

and the timing is

$t' = t+\#L \iff k=\#L \ then \ ok$

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:

$J \wedge k = \#L \wedge \text{ok}$

in context $k = \#L \wedge p' = p$ the first conjunct of $J$ is $P$

$\Rightarrow P$

Second case:

$J \wedge k+\#L \wedge L \ k \geq 0$

we need $J \wedge k+\#L \wedge L \ k \geq 0$ 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 $L \ k$ is nonnegative, the maximum product of these

extensions is the maximum product $c$ of those sequences ending at $k$ times the

new item $L \ k$. So the maximum product of all segments ending at or before $k$ is

the maximum of $p$, $c \times L \ k$, and 1. That's what the first two conjuncts of the

maximum product of any segment (sublist of consecutive items).

The problem is

$P$ defined as

$P = p' = {\prod}_{i:0..L+1} \cdot {\prod}_{j: i..L+1} \Pi \ L \ [i..j]$ 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

$J = p = {\prod}_{i:0..k+1} \cdot {\prod}_{j: i..k+1} \Pi \ L \ [i..j]$

and the timing is

$t' = t+\#L \iff p := 1 \cdot c:= 1 \cdot d:= 1 \cdot k:= 0 \cdot t' = t+\#L-k$

and the timing is

$t' = t+\#L-k \iff k=\#L \ then \ ok$

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:

$J \wedge k = \#L \wedge \text{ok}$

in context $k = \#L \wedge p' = p$ the first conjunct of $J$ is $P$

$\Rightarrow P$

Second case:

$J \wedge k+\#L \wedge L \ k \geq 0$

and the timing is

$t' = t+\#L \iff k=\#L \ then \ ok$

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:

$J \wedge k = \#L \wedge \text{ok}$

in context $k = \#L \wedge p' = p$ the first conjunct of $J$ is $P$

$\Rightarrow P$

Second case:

$J \wedge k+\#L \wedge L \ k \geq 0$

and the timing is

$t' = t+\#L \iff k=\#L \ then \ ok$

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:

$J \wedge k = \#L \wedge \text{ok}$

in context $k = \#L \wedge p' = p$ the first conjunct of $J$ is $P$

$\Rightarrow P$

Second case:

$J \wedge k+\#L \wedge L \ k \geq 0$

and the timing is

$t' = t+\#L \iff k=\#L \ then \ ok$
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.