212 (next sorted list) Given a nonempty sorted list of naturals, write a program to find the next (in list order) sorted list having the same length and sum.

After trying the question, scroll down to the solution.

§ Here are 3 sorted lists, and beside each is the sorted list with the same length and sum that comes next in list order.

[1; 4; 10] → [1; 5; 9]
[1; 1; 2; 3] → [1; 2; 2; 2]
[1; 1; 1; 2] → none

The list [1; 1; 1; 2] has no sorted list with the same length and sum that comes later in list order. The question does not say what the result should be in such cases.

Let L be the list variable. Define *sorted* as follows.

$$sorted\ L = \forall i, j: 0,..\#L \cdot\ i \le j \Rightarrow L\ i \le L\ j$$

The problem specification is

$$L: [*nat] \wedge \#L>0 \wedge sorted\ L$$
$$\Rightarrow\quad L': [*nat] \wedge \#L'=\#L \wedge \Sigma L'=\Sigma L \wedge sorted\ L' \wedge L'>L$$
$$\wedge\ \neg\exists M: [*nat] \cdot\ L<M<L' \wedge \#M=\#L \wedge \Sigma M=\Sigma L \wedge sorted\ M$$

Unfortunately that's unimplementable because there may not be any such L'. So define *between L N* to means there is a sorted list with the same length and sum as L that comes between L and N in list order.

$$between\ L\ N = \exists M: [*nat] \cdot\ L<M<N \wedge \#M=\#L \wedge \Sigma M=\Sigma L \wedge sorted\ M$$

Now the implementable specification S is defined as

$$L: [*nat] \wedge \#L>0 \wedge sorted\ L \wedge between\ L\ [\#L*\infty]$$
$$\Rightarrow L': [*nat] \wedge \#L'=\#L \wedge \Sigma L'=\Sigma L \wedge sorted\ L' \wedge L'>L \wedge \neg between\ L\ L'$$

Let i and j be natural. To ensure the length and sum stay the same, the only change to list L will be

$$L:=\ i \to L\ i+1 \mid j \to L\ j-1 \mid L$$

when $0 \le i<\#L \wedge 0 \le j<\#L \wedge i \ne j$. To ensure the list changes to a later list in list order, we need $i<j$. Putting these conditions together, we need $0 \le i<j<\#L$. The list is initially sorted, and we want the new list to be sorted, so we need

$$L\ i+1 \le L(i+1) \wedge L(j-1) \le L\ j-1$$

which can be simplified to

$$L\ i < L(i+1) \wedge L(j-1) < L\ j$$

All together, we need

$$0 \le i<j<\#L \wedge L\ i < L(i+1) \wedge L(j-1) < L\ j$$

What remains is to find the next such list; for that, we want i and j to be as large as possible. Informally:

(find largest j such that $0<j<\#L \wedge L(j-1) < L\ j$).
(find largest i such that $0 \le i<j \wedge L\ i < L(i+1)$).
$L:=\ i \to L\ i+1 \mid j \to L\ j-1 \mid L$

Formally, define

$$findj = j' = \Uparrow k: (\S k: 1,..\#L \cdot\ L(k-1) < L\ k) \cdot\ k$$
$$findi = i' = \Uparrow k: (\S k: 0,..j \cdot\ L\ k < L(k+1)) \cdot\ k$$
$$assignL = L:=\ i \to L\ i+1 \mid j \to L\ j-1 \mid L$$

In *findj*, if $\S k: 1,..\#L \cdot\ L(k-1) < L\ k$ is empty, then $j'=-\infty$.
In *findi*, if $\S k: 0,..k \cdot\ L\ k < L(k+1)$ is empty, then $i'=-\infty$.
So we change i and j to be extended integers. Now the specification is

findj.
if $j \ne -\infty$ **then** $j'=j \wedge$ *findi*.
if $i \ne -\infty$ **then** *assignL*
else *ok* **fi**
else *ok* **fi**

I should now prove that $S \Leftarrow$(the above specification) but it looks long and hard and I'm too lazy. To find the largest index, we search from the end toward the beginning of the list.

Define *newj* $=$ $j' = \Uparrow k$: $(\S k$: $1,..j \cdot L(k-1) < L\, k) \cdot k$

Then *findj* $\Leftarrow$ $j := \#L.$ *newj*
and *newj* $\Leftarrow$ $j := j-1.$ **if** $j=0$ **then** $j := -\infty$ **else if** $L(j-1) < L\, j$ **then** *ok* **else** *newj* **fi fi**

Define *newi* $=$ $i' = \Uparrow k$: $(\S k$: $0,..i \cdot L\, k < L(k+1)) \cdot k$

Then *findi* $\Leftarrow$ $i := j.$ *newi*
and *newi* $\Leftarrow$ $i := i-1.$ **if** $L\, i < L(i+1)$ **then** *ok* **else if** $i=0$ **then** $i := -\infty$ **else** *newi* **fi fi**

The proofs of the *findj* and *findi* refinements are each one use of the Substitution Law. The proofs of the *newj* and *newi* refinements look easy enough but I'm still too lazy.

With thanks to Łukasz Jakimczuk.