213 (next combination) You are given a sorted list of m different numbers, all in the range $0,..n$. Write a program to find the lexicographically next sorted list of m different numbers, all in the range $0,..n$.

After trying the question, scroll down to the solution.

§ Here is the last sorted list of 5 different numbers all in the range 0,..10 .

[5; 6; 7; 8; 9]

At index i , the largest possible item is $n–m+i$. Strategy: find the last item that is below its maximum, increase it by 1 , then fill up the following items in increasing order. For example, if the sorted list of 5 different numbers in the range 0,..10 is

[2; 4; 7; 8; 9]

the last item that is below its maximum is the 4 . So increase the 4 to 5, then fill up the rest and get

[2; 5; 6; 7; 8]

To find the last, we can search from the end back toward the beginning. To make the specification implementable, we have to decide what to do if we are given the last list; I choose that we leave it as is.

Let L be a list variable whose initial value is the given sorted list of length m with items all in $0,..n$. Here are some facts about L .

L: $[m*(0,..n)]$
$\forall j, k: \Box L\cdot\ j<k \Rightarrow L\,j < L\,k$
$\forall j: \Box L\cdot\ L\,j \le n–m+j$
$\forall j: \Box L\cdot\ L\,j < n–m+j \Rightarrow \forall k: 0,..j\cdot\ L\,k < n–m+k$
$\forall j: \Box L\cdot\ L\,j = n–m+j \Rightarrow \forall k: j,..m\cdot\ L\,k = n–m+k$

Let F be the conjunction of the above facts. Let i be a *nat* variable used to index L . Define specifications

S = **if** $L = [n–m;..n]$ **then** $L'=L$ **else** $F' \wedge L'>L$ **fi**
A = find last item below its max at index $i'–1$ in $0,..i$
= $F \Rightarrow L'=L \wedge (i'=0 \vee L(i'–1) < n–m+i'–1) \wedge \forall k: i',..i\cdot\ L\,k = n–m+k$
B = fill up increasing items at $i,..m$
= $(\forall k: 0,..i\cdot\ L'k = L\,k) \wedge (\forall k: i,..m\cdot\ L'k = L(k–1)+1)$

The refinements are

$S \Leftarrow$ $i:= m.\ A.$ **if** $i=0$ **then** *ok* **else** $L:= i–1 \rightarrow L(i–1) + 1 \mid L.\ B$ **fi**

$A \Leftarrow$ **if** $i=0$ **then** *ok* the given list is the last
else if $L(i–1) = n–m+i–1$ **then** $i:= i–1.\ A$ the item is max
else *ok* **fi fi** item i is the last below its max

$B \Leftarrow$ $i:= i+1.$ **if** $i=m$ **then** *ok* **else** $L:= i \rightarrow L(i–1)+1 \mid L.\ B$ **fi**

The proofs are UNFINISHED.

I thank Łukasz Jakimczuk for pointing out that if the last item that is below maximum is increased, and if that makes it maximum, then we don't need to update the following items. He also pointed out that binary search works.