You are given an unsorted list of length $n$ whose items are the numbers $0..n+1$ with one number missing. Write a program to find the missing number.

Let the given list be $L$ (a constant), and its length is $n$ (a constant). Then

$L(0..n) : 0..n+1 \land \forall i,j: 0..n \cdot i + j \Rightarrow L i + L j$

Let $m : \text{nat}$ be a variable whose final value will be the missing number. The problem can be stated

$m' : 0..n+1 \land \neg m' : L(0..n)$

One way to solve the problem is with an extra list variable $M : [(n+1)\ast \text{bin}]$ to record which numbers are present in $L$. Here's an easier way: the problem is

$m' = (\Sigma[0..n+1]) - (\Sigma L)$

and its solution is

\[
m' = (\Sigma[0..n+1]) - (\Sigma L) \iff m := n \ast (n+1)/2.\ A 0 \Rightarrow A'n \\ A 0 \Rightarrow A'n \iff \text{for } i := 0..n \text{ do } i : 0..n \land A i \Rightarrow A'(i+1) \text{ od}
\]

\[i : 0..n \land A i \Rightarrow A'(i+1) \iff m := m - L i\]

where invariant $A i = m = (\Sigma[0..n+1]) - (\Sigma L[0..i])$.

Similarly the problem can be stated as

$m' = n + \Sigma i : 0..n \cdot i - L i$

and solved as

\[
m' = n + \Sigma i : 0..n \cdot i - L i \iff m := n.\ B 0 \Rightarrow B'n \\ B 0 \Rightarrow B'n \iff \text{for } j := 0..n \text{ do } j : 0..n \land B j \Rightarrow B'(j+1) \text{ od}
\]

\[j : 0..n \land B j \Rightarrow B'(j+1) \iff m := m + j - L j\]

where invariant $B j = m = n + \Sigma i : 0..j \cdot i - L i$. For either solution, recursive time is $i' = t+n$. 