Exercise sheet 9

Exercise 1 (Insertion sort).

Prove `forall l, Permutation (isort l) l` where `isort` is the insertion sort function from the lecture.

***Exercise 2** (Sortedness).

Prove

```
forall l,
   (forall i j, i < j < List.length l -> List.nth i l 0 <= List.nth j l 0) ->
   Sorted l.
```

where `Sorted` is the inductive predicate from the lecture expressing that a list of natural numbers is sorted.

***Exercise 3** (Fibonacci numbers).

Recall the `fib` and `fib'` functions from the first lecture:

```
Fixpoint fib (n : nat) : nat :=
  match n with
  | 0 => 1
  | 1 => 1
  | S (S m as m') => fib m + fib m'
  end.

Fixpoint itfib a b k :=
  match k with
  | 0 => a
  | S m => itfib b (a + b) m
  end.

Definition fib' n := itfib 1 1 n.
```

The functions compute the nth Fibonacci number, in time exponential in n and linear[1] in n, respectively.

Prove: `forall n, fib n = fib' n`.

[1]Assuming addition on `nat` is a basic operation, i.e., counting it as constant time.