Challenge

Delegate A/S

November 12, 2015

Introduction

Many challenges use lists with an added time constraint, such as "do X using only n recursive calls". In some of these challenges append would be very useful, however append itself runs in $O(n)$.

Problem

Implement, in F#, a small module in which append runs in $O(1)$, without using lazy.

```
module FastList =
  type 'a fast_list = ...
  let nil = ...
  let cons x xs = ...
  let rec append xs ys = ...
  let rec iter f xs = ...

open FastList
let test1 = cons 45 (cons 44 (cons 43 (cons 42 nil)))
let test2 = cons 4 (cons 3 (cons 2 (cons 1 nil)))

append test1 test2 |> iter (printfn "%i")
```