More Inductive Types
Last Lecture

- You can figure out "insert sort" yourself
  - If you follow the recipe, carefully
  - Recipe tells you:
    - Think about base case
    - Think about recursive case (and what to do with result of recursive call)
    - Make a wish-list (e.g. "insert"), and repeat!

- Remember this for current homework
Today's Lecture

- A short-hand for lists
- Another inductive type: trees!
List Shorthands

- \((\text{cons } 1 (\text{cons } 2 (\text{cons } 3 \text{ empty})))\)
- \(= (\text{list } 1 2 3)\)
- \((\text{cons } (\text{cons } 1 (\text{cons } 2 (\text{cons } 3 \text{ empty})))\) empty\)
- \(= (\text{list } (\text{list } 1 2 3))\)
- \((\text{cons } (\text{list } 1 2) (\text{list } 3 4))\)
- \(= (\text{list } (\text{list } 1 2) 3 4)\)
A Type for Family Trees

; A family-tree-node (short: ftn) is either
; - empty; or
; - (make-child f m na da ec)
; where f and m are ftns, na
; and ec are symbols, and da is a
; number.
Blue-eyed Ancestor

; blue-eyed-ancestor? : ftn -> boolean
; to determine whether
; a-ftree contains a child
; structure with 'blue in the
; eyes field