



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



List Shorthands

- `(cons 1 (cons 2 (cons 3 empty)))`
- `= (list 1 2 3)`
- `(cons (cons 1 (cons 2 (cons 3 empty)))
empty)`
- `= (list (list 1 2 3))`
- `(cons (list 1 2) (list 3 4))`
- `= (list (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
```