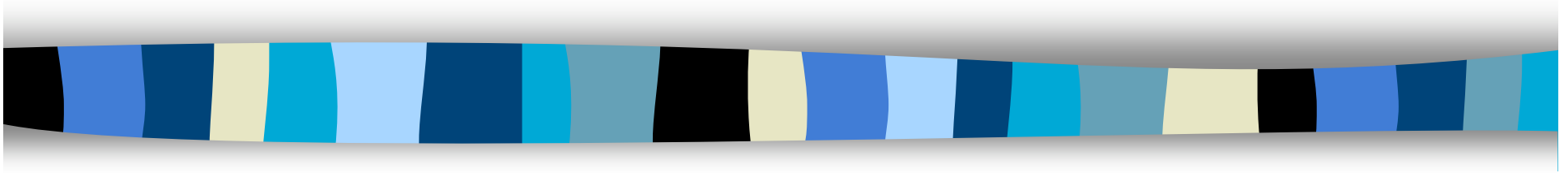


More on Lists



Problems that need auxiliary
functions that process lists



Today's Goals

- List abbreviations
- Bigger functions that process lists
- Example: Sort
- Example: Arrangements (permutations)



List Abbreviations

- We use lists a lot when we program
- Using empty and cons is verbose
- Example:
 - (list 1 2 3)
 - (cons 1 (cons 2 (cons 3 empty)))
- Example
 - '((1 2) (3 4))
 - (list (list 1 2) (list 3 4))
 - (cons (cons 1 (cons 2 empty)) (cons (cons 3 (cons 4 empty)) empty))



Functional Decomposition

- A technique for program analysis
- Look for what should be a function
- Sign: A dependency

- Goal:
 - To divide problem into smaller part
 - Not to solve the smaller parts (yet)



Example: Insert Sort

- Sort: [number] -> [number]
- Example:
 - Given '(3 5 4 2 1)
 - Return '(1 2 3 4 5)
- Example:
 - Given '(100 3 400)
 - Return '(3 100 400)
- Can we devise a method for doing this?



Example: Arrangements

- Arrange: [symbol] -> [[symbol]]
- Example:
 - Given '(a b)
 - Return '((a b) (a b))
- Example:
 - Given '(a b c)
 - Return '((a b c) (c a b) (b c a) (c b a) (a c b) (b a c))



For Next Class

- Homework due Monday
- Reading:
 - Chapters 14 + companion notes
 - Trees!
- Quiz:
 - Chapter 14