CS3L:
Introduction to Symbolic Programming

Lecture 9:
Recursion Rocks!

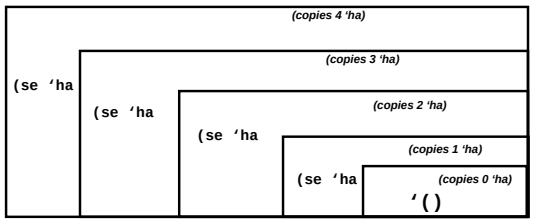
Summer 2008 Colleen Lewis
colleenL@berkeley.edu

Today

- Copies (we need volunteers)
- Count-evens
- Find-evens
- Recursion In Two Directions

Copies

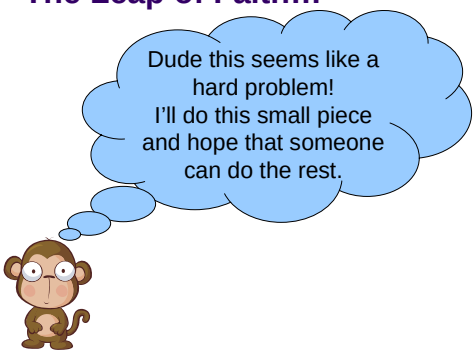
```
(define (copies n wd)
  (if (< n 1)
      '()
      (sentence wd (copies (- n 1) wd))))
```



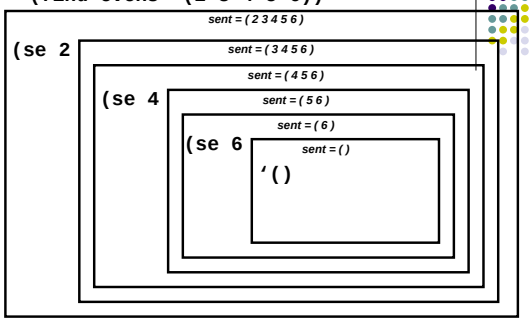
Count the number of even-numbers

```
(define (count-evens sent)

  (cond ((empty? sent)  ;empty?
         0 )            ;base case: return 0

        ((even? (first sent))
         (+ 1
           (count-evens (bf sent))));recurse on the
                                    ; rest of sent

        ((odd? (first sent)
         (+ 0
           (count-evens (bf sent))) ;recurse on the
                                    ; rest of sent

   ))
```

The Leap of Faith…




```
> (find-evens '(2 3 4 5 6))
```

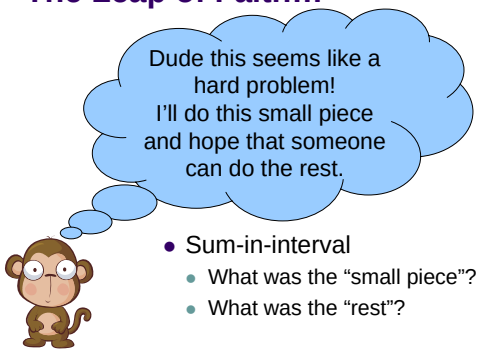



```
➔ (se 2 (se 4 (se 6 '())))
➔ (2 4 6)
```

Problem: ***find all the even numbers in a sentence of numbers***

```
(define (find-evens sent)
  (cond ((empty? sent)       ;base case
         '()       )
        ((odd? (first sent)) ;rec case 1
         (find-evens (bf sent)) )
        (else                ;rec case 2: even
         (se (first sent)
             (find-evens (bf sent))) )
        ))
```

The Leap of Faith…



- Sum-in-interval
  - What was the "small piece"?
  - What was the "rest"?

Sum-In-Interval