Seat Assignments for Today

Jenny and Brody at **Meltzer**

David R. and Noel at **Karpinski**

Lexy and Eujin at **Licklider**

Merel and Will at **Hermann**

Zikai and Mai at **Papert**

Kei and Cynthia at **Holberton**

Eamon and David H. at **Williamson**

Luoyu and Luke at **Keonig**

Ziwei and Aiden at **Grinbergs**

Yun and Bjorn at **Rhodes**

Hallie and Clare at **Keller**

Dean and Amy at **Easley**

CSC 151 Spring 2020
Topic: Basic Recursion

February 28, 2020

Day 17

Agenda for today

- Peer Support Opportunities
- Quiz 5 topics
- Recursion Basics!
- Lab
- Wrap-up & Feedback

Peer Support Opportunities

Indoor Track Meet, Friday 1pm and Saturday 9:30am

Quiz 5 Topics

No quiz 5!

Parts of Algorithms

What are the parts of algorithms we discussed at the start of the course? How are these parts available to us in Scheme?

Discuss with your partner(s).

Two Parts of a Recursive Procedure

Base Case

The problem is so small we can solve it right away. Do so.

Recursive Case

Solve a small part of the problem

Recursively solve the rest of the problem using the same procedure

Combine the results

How do you paint the layers of a Matryoshka (Russian Nesting Doll)?



How do you paint the layers of a Matryoshka (Russian Nesting Doll)?

Base Case

If the doll cannot be divided, you should paint it.

Recursive Case

If it can be divided, then paint the inner doll, then paint both halves of the current doll.

How do you paint the layers of a Matryoshka (Russian Nesting Doll)?

Psuedo-code:

```
(define paint-matryoshka
  (lambda (doll)
     (if  (doll cannot be divided) (paint doll)
        ( (paint-matryoshka (doll – layer)) then paint current layer)))
```

Unwrapping a present

Your friend has given you a birthday present. Your friend has placed it in a box and then wrapped it in several layers of wrapping paper. You must develop a recursive algorithm for opening the present.

Base Case

Recursive Case

Unwrapping a present

Your friend has given you a birthday present. Your friend has placed it in a box and then wrapped it in several layers of wrapping paper. You must develop a recursive algorithm for opening the present.

Base Case

If **t**here is no more wrapping paper, you should open the box.

Recursive Case

If there is a layer of wrapping paper, take off a layer of wrapping paper, then open the inner layer.

Unwrapping a present

Psuedo-code for Open a Present:

```
(define open-present
  (lambda (present)
    (if
                          )))
```

Unwrapping a present

Psuedo-code for Open a Present:

```
(define open-present
  (lambda (present)
    (if  (present has no layer of wrapping paper) (open box)
       (open-present (present – layer of paper))))
```

Summing Numbers 1 through 6

I need Volunteers

Summing Numbers in a list

Formally: suppose we have a list of numbers. Write a recursive algorithm to sum up the elements of the list.

What is the base case?

What is the recursive case?

Summing Numbers in a list

Formally, suppose we have a list of numbers. Write a recursive algorithm to sum up the elements of the list.

What is the base case?

If the list is empty, return 0

What is the recursive case?

If the list is not empty, keep the first element, and sum it to the rest of the numbers.

Summing Numbers in a list

Formally, suppose we have a list of numbers. Write a recursive algorithm to sum up the elements of the list.

What is the Scheme-code?

```
(define sum-numbers
   (lambda (lst)
     (if
                                    ))))
```

Summing Numbers in a list

Formally, suppose we have a list of numbers. Write a recursive algorithm to sum up the elements of the list.

What is the Scheme-code?

```
(define sum-numbers
   (lambda (lst)
     (if  (null? lst) 0
         (+ (car lst) (sum-numbers (cdr lst)) ))))
```

Lab Time

Start with the person on the East side of the room.

Upcoming Work

We will continue this lab on Monday.

Assignment 5 will be posted by 5pm today

Mid Semester Evaluation of CSC-151