# CSC 172– Data Structures and Algorithms

Lecture #13 Spring 2018

Please put away all electronic devices



## Announcement

- Midterm on this Thursday
  - Same location and time as for regular lectures
  - Time: 75 minutes



• Recursion

• Sorting

# **Recursion Versus Iteration**

Recursion and iteration are similar

#### Iteration

A loop repetition condition determines whether to repeat the loop body or exit from the loop

#### Recursion

the condition usually tests for a base case

You can always write an iterative solution to a problem that is solvable by recursion

A recursive algorithm may be simpler than an iterative algorithm and thus easier to write, code, debug, and read

#### BUT

### **Tail Recursion or Last-Line Recursion**

- When recursion involves single call that is at the end ...
- It is called **tail recursion** and it easy to make iterative

```
int factorial(int n) {
    if (n == 0)
        return 1;
    else
        return n * factorial(n - 1);
}
```

It is a straightforward process to turn such a function into an iterative one

### Iterative factorial function

```
int factorial_iter(int n) {
    int result = 1;
    for (int k = 1; k <= n; k++)
    result = result * k;
    return result;
}</pre>
```

# **REAL WORLD EXAMPLE**

# **Towers of Hanoi**

- Move the three disks to a different peg, maintaining their order (largest disk on bottom, smallest on top, etc.)
  - Only the top disk on a peg can be moved to another peg
  - A larger disk cannot be placed on top of a smaller disk



# Let's Play

<u>http://www.mathsisfun.com/games/towerofhan</u>
 <u>oi.html</u>

## **Recursive Algorithm for Towers of Hanoi**

#### **if** *n* is 1

Move disk 1 (the smallest disk) from the starting peg to the destination peg

#### else

- Move the top n 1 disks from the starting peg to the temporary peg (neither starting nor destination peg)
- Move disk *n* (the disk at the bottom) from the starting peg to the destination peg
- Move the top n 1 disks from the temporary peg to the destination peg

# BACKTRACKING

## Solving a Maze



Backtracking:

*Systematic* trial and error search for solution to a problem

# Backtracking





http://eightqueen.becher-sundstroem.de/

# Acknowledgement

- Douglas Wilhelm Harder.
  - Thanks for making an excellent set of slides for ECE
     250 Algorithms and Data Structures course
- Prof. Hung Q. Ngo:
  - Thanks for those beautiful slides created for CSC 250 (Data Structures) course at UB.
- Many of these slides are taken from these two sources.

### **After Midterm**

• Sorting and Sorting Dance



# **MIDTERM REVIEW**

# **MIDTERM REVIEW**

## **Chapters to Read**

- Chapter 1
- Chapter 3 (Mathematical Background)
- Chapter 4 (Algorithm Analysis)
- Chapter 5 (Linear Structures)
- Chapter 6 (you can skip)
- Chapter 7 (Recursion)

## **Topics Covered**

- Java Generics
- Induction
- Program Efficiency
- Asymptotic Analysis
- Recurrence Tree
- ArrayList and LinkedList
- Stack (+ Shunting Yard Algorithm)
- Queue
- Recursion

What did we learn?

# WHAT DID WE LEARN?

## DAG



# DAG (Continued)



Now, you know how to create your own data structures

# DAG (Continued)



# **DATA STRUCTURES COVERED**

## **Linear Data Structures Covered**

- Arrays
- ArrayLists
- LinkedLists
- Stacks
- Queues

# After the MT: Other Data Structures

- Trees
  - LMC-RS Trees
  - Binary Trees
    - Binary Search Trees
      - Balanced Binary Search Trees (AVL trees)
        - » TreeMaps (Map implementation using Balanced Binary Search Tree) •
    - Heaps (and Priority Queues)
- Graphs
- HashMaps
- Sets?

# **ALGORITHMS**

# **Algorithms**

- Linear search vs Binary search
- Finding well-formed expression
- Shunting yard algorithm
- Recursion (and backtracking)



### **Projects and Data Structures**



# **EXAM STRUCTURE**

### **Exam Structure**

Total: 60 pts Time: 75 Minutes Average: 1 minute per point Coverage: Everything including Recursion

#### • Short Questions (20 pts):

- One sentence answers
- Short questions
- Short functions

#### • Other Questions (40 pts):

- Questions will have parts
- All: Coding, Drawing, Theory,
- Also, there will be extra credit problem

## How to prepare for the exam

• Go through workshop materials, labs, quizzes and above all, lecture slides

- Read the textbook to strengthen your understanding
  - Bonus: plenty of animations

# **THINGS TO AVOID**

# The Night before exam



## During the exam



# Or Finishing too early



WeKnowMemes

# **Practice Problems**

• Workshop 6 (already uploaded)

- Lectures
- Quizzes
- Labs
- Workshops
- Projects

If you have attended all lectures, have completed all labs and projects on you own, it will be <u>hard</u> not to do well in the exam!

# Coding

- Generics
- Recursion
- Iteration vs Recursion
- Implementing ADT
  - ArrayList
  - LinkedList
  - Stack
  - Queue
- Using Arrays, ArrrayList, LinkedList, Stack, Queue
- Using Singly Linked List



## **MT** Evaluation

- Time for MT Evaluation:
- Dr. Jenny Hadingham
  - Assistant Director
  - <u>Center for Excellence in Teaching & Learning</u>