Exam 1 Review

TOPICS: What you need to know

1. Two key notions: Correctness & Efficiency
2. Computational problems: How to write a well-defined problem specification
3. Computational problems: estimating inherent complexity (Ω notation) from the problem specification
4. Coming up with more than one computational strategy for solving a problem
    1. The notion of problem decomposition
    2. Recursive strategies
    3. Iterative strategies
    4. Thinking about correctness and efficiency of strategies
5. Turning strategies into algorithms
    1. Recursive algorithms
    2. Divide & conquer recursive algorithms
    3. Avoiding duplicated work
    4. Avoiding tail recursion
    5. Iterative algorithms
    6. Divide & conquer iterative algorithms
    7. Removing recursion
6. Reading, Writing and Understanding algorithms
    2. The language of algorithms: Pseudocode
    3. Understanding the mechanics of recursive algorithms: Recursion Trees of the algorithm solving a problem instance
    4. Understanding the mechanics of non-recursive algorithms: by working out the loops etc. of the algorithm solving a problem instance
7. Hypothesizing and proving correctness/incorrectness of algorithms
    1. Proof by counterexample (incorrectness proof)

TECHNICAL SKILLS: What you should be able to do

- Specify a problem so that it is well-defined with the three components of a well-defined problem explicitly stated
- Estimate the inherent complexity of a problem
- Develop computational strategies to solve a problem
- Translate strategies into algorithms
- Design algorithms using the design techniques of recursion, iteration and divide-and-conquer
- Write algorithms in pseudocode
- Understand and explain the mechanics of algorithms
    - Mental simulation
    - Explaining the operation of iterative algorithms
    - Explaining the operation of recursive algorithms (Recursion Tree)
- Make an informed determination of algorithm correctness
    - Check for boundary conditions of inputs, loops and recursion
- Prove algorithm incorrectness using Proof by Counterexample

How to Prepare

1. Review lecture slides and any notes you took in class
2. Read the assigned readings from the text:
    1. All of Chapter 1
    2. Chapter 2 p. 20-23
    3. Chapter 2
        1. Section 2.1: Omit (for the time being) the discussion of loop invariants (p. 18-20); read the rest
        2. Section 2.3: Omit (for the time being) Section 2.3.2 and the discussion of loop invariants (p. 32-33) ; read the rest
    4. Chapter 32 p. 988-989
3. Refresh your knowledge about sorting algorithms from COMP 2210: Selection & Bubble in addition to those discussed in class: Insertion and Merge
4. Review homework solutions and ensure that you are able to solve similar problems
5. Work out thinking assignments from the slides with your friends

Exam Structure

- Take-home exam handled electronically via Canvas
- 5 Problems 25 multiple choice questions 50 points
- Exam is open text and notes
- All electronic devices are allowed
- It is prepared like a 50 minutes in-class exam
- No need to memorize anything. Any mathematical results or algorithms you need will be provided with the exam.

- **The class on Tuesday September 22 will be reserved for taking the exam at home**. No lecture on September 22.

- IT IS HIGLY RECOMMENDED THAT YOU COMPLETE THE EXAM WITHIN 50 MINUTES.

Added Flexibility - time and a half

- Time and a half on Exam I for everyone.

- Available at 9:30 AM Tuesday September 22
- Due before 10:45 AM Tuesday September 22

- For students needing **accommodation**, this will allow sufficient time for taking the Exam and its take-home nature automatically addresses your needs.

- Submission deadline will be strictly enforced.

- **Late submission will not be accepted**.

Academic Honesty

IF I SEE YOU ENGAGE IN ANY KIND OF CHEATING OR SEE EVIDENCE IN YOUR ANSWERS, YOU WILL FAIL THE COURSE AND THE CASE WILL BE REPORTED