COSC-311 Fall 2018 Midterm 1 Topics

This is *not* a comprehensive study guide. There may be topics that we have discussed in class or that have come up on homework that are not on this list. You are responsible for all of the course material up to this point, including both in-class material and homework.

1. Sorting algorithms

   (a) The algorithms we've seen:

   - Selection sort
   - Heapsort
     - Heap properties
     - Linear-time heapify
   - Mergesort
   - Quicksort
   - Insertion sort (on HW1)
   - Bubble sort (on HW1)

   (b) For each algorithm, know:

   - How the algorithm works (i.e., be able to run it by hand on an array)
   - Its asymptotic runtime and where the runtime analysis comes from

2. Divide-and-Conquer paradigm

   (a) Three components to a divide-and-conquer algorithm

   (b) How to write a recurrence for the runtime of a divide-and-conquer algorithm

3. Theoretical topics

   (a) Asymptotic analysis

   - What is it?
   - Best case vs. worst case analysis
   - Definitions of Big-O, Big-Ω, Θ
   - How to show a function is in $O(g(n))$ (or Ω, or Θ)
   - How to determine the asymptotic runtime of an algorithm

   (b) Proof by induction: what are the steps to an inductive proof, and why do those steps form a sufficiently convincing proof?

   (c) Solving recurrences

   - Via recursion trees
   - Via induction

FAQ

Q: What's the level of difficulty of the exam questions, relative to homework?

A: The exam questions will be closer to Mini HW than to HW. You have two weeks to do each homework assignment, and I expect that you're spending lots of time thinking deeply about the problems. You have 50 minutes to do the exam, and so the questions that I ask won't be as in-depth as homework questions.

Q: But what will the actual questions look like?

A: Here are some types of questions that I might ask (this is not a comprehensive list):

- Given a new algorithm that you haven't seen before, analyze its worst-case runtime.
- Given an input array and a particular sorting algorithm, run the algorithm on that input.
- Show that a particular function is in Big-Ω of another function.
- Given a target application, discuss which sorting algorithm you would use and why.
- Given a new problem, how would you adapt or modify an algorithm we've discussed to solve the new problem?

Q: How should I go about studying?

A: The most important recommendation I can give is to study actively. Don't simply read your notes and then conclude that you've mastered the material. Instead, go back and redo previous mini homeworks. Re-derive the solutions to examples we did in class without looking at your notes. Make up some functions and prove asymptotic bounds for them. Make up some unsorted arrays and sort them using different algorithms.

Q: You're going to be out of town. What if I have a question during the exam?

A: I've tried to write the exam so that the questions are unambiguous about what I am asking you to do. However, if you're confused about what you're supposed to do, you can write down the question you would like to ask me and the answer that you think I'd give. If you've misinterpreted what a question means, I will take this into account when grading.

Q: Will the exam be curved?

A: I aim to write my exams so that the median falls somewhere around a B+ (i.e., in the high 80s), and I do not intend to curve the scores. However, I reserve the right to change my mind should I discover after the fact that I've substantially miscalibrated the difficulty of the exam.