csce350 — Data Structures and Algorithms
Fall 2019 — Review Sheet for Test 2

Test date: Tuesday, November 5, in class

This review sheet is a general outline of the topics we covered in this class.

- ✓ The test will cover Chapters 4, 5, and 6 of the textbook. This includes includes classes from September 24 through October 29.
- ✓ You'll have 75 minutes to complete the test.
- ✓ Notes are permitted subject to the following conditions, about which I am quite serious.
  - The notes must be on a single note card measuring 3 inches by 5 inches or less.
  - Yes, you may write on both sides.
  - You must write the notes yourself, using your hand and a pencil or pen.
  - You must write your name in the top left corner.
  - You must submit your note card with your test, but I will return it later at your request.
  - You must not make any modifications that change the surface area of the card, nor use magnifiers, colored films, nor any other objects to assist with reading the card.
  - If you prefer, you may choose not to use any notes at all.
- ✓ No calculators nor other reference material are permitted.
- ✓ The test packet will include copies of pages 475, 476, and 477 of the textbook. No other reference material will be included.
- ✓ Possible question types include, but are not limited to:
  - Problems created by mutation of the homework. (. . . but be prepared to think!)
  - Problems similar in spirit to the homework, but related to different concepts from the lectures.
  - Questions evaluating your mastery of the new terminology and concepts introduced in the course. Example formats: Multiple choice, fill-in-the-blank, short answer.

General

1. For each algorithm we covered, be prepared to: Explain the problem it solves, including the inputs and outputs. Describe the basic idea of how the algorithm works, and why each part of its pseudocode is important. Classify the underlying algorithm design strategy. Execute the algorithm on appropriately-sized inputs. Analyze the run time of the algorithm.

Chapters 1 and 2

1. Not covered directly, but you should be prepared to design and analyze algorithms based on the techniques from Chapters 4–6.

Chapter 4 (Decrease and Conquer)

1. Definition of decrease and conquer.
2. Integer powers, decrease and conquer version.
3. Insertion sort
4. Binary search
5. QuickSelect ("Hoare's selection algorithm") What is the role of the partition function? How does partitioning work? How does the choice of pivot impact the algorithm's run time?
6. Not covering: topological sorting, generating permutations, generating subsets, fake-coin problem, Russian peasant multiplication, Josephus problem, interpolation search, game of Nim

Chapter 5 (Divide and Conquer)

1. Definition of divide and conquer. How does it differ from decrease and conquer?
2. Master theorem. Be able to identify recurrences to which the Theorem applies. Be able to simplify the answers where appropriate.
3. Mergesort. What is the role of the merge function?
4. Quicksort. What is the role of the partition function? How does the choice of pivot impact the algorithm's run time?
5. Multiplication of large integers. Brute force ("pencil-and-paper") algorithm. Karatsuba algorithm.
6. Not covering: binary tree traversals, Strassen's algorithm, closest pair, convex hull

Chapter 6 (Transform and Conquer)

1. Definition of transform and conquer.
2. Presorting. When is this a good idea? Why?
3. AVL trees. Background: Inserting and search in standard binary search trees. An AVL tree *is* a binary search tree. What problem with BSTs do AVL trees solve? Definition of balance factor. What balance factors are allowed in an AVL tree? Rotations: Left and right. Where do the subtrees go during a rotation? Algorithm to rebalance a tree. How many rotations are needed in the worst case? **Practice problem for AVL trees: 6.3.4**
4. Skip for now: 2-3 Trees. Heaps. Gaussian elimination, Horner's rule, binary exponentiation, problem reductions.