CS 5633 Analysis of Algorithms – Fall 16

Exam 1

NAME:

- This exam is closed-book and closed-notes, and electronic devices such as calculators or computers are not allowed. You are allowed to use a cheat sheet (half a single-sided letter paper).
- Please try to write legibly – if I cannot read it you may not get credit.
- **Do not waste time** – if you cannot solve a question immediately, skip it and return to it later.

1) Big-Oh Notation		10
2) Recursion Tree		15
3) Induction		20
4) Master Method		15
5) Randomized Analysis		15
6) Divide and Conquer		25
		100

1 Big-Oh Notation (10 Points)

Use the definition of Big-Oh and Omega to show that $4n^2 - 8n + 6 \in \Theta(n^3)$.

2 Recursion Tree (15 Points)

Use a recursion tree to generate a guess of the asymptotic complexity of a divide and conquer algorithm with the running time $T(n) = 4T(n/2) + O(n^3)$.

3 Induction (20 Points)

Let $T(n) = 3T(n/3) + dn$ with $T(1) = a$ for constants d and a. Use induction to prove that $T(n) \in O(n \log_3 n)$.

4 Master Method (15 Points)

Solve the following recurrences using the master theorem. Justify your answers shortly (i.e. specify ϵ and check the regularity condition if necessary).

1. $T(n) = 2T(n/2) + n^2$

2. $T(n) = 27T(n/3) + n^3$

3. $T(n) = 9T(n/3) + n \log n$

5 Randomized Analysis (15 Points)

Suppose someone offers to let you play a game. They will randomly (and independently) roll three dice: a red die, a green die, and a blue die. If you choose to play, you will pay $5 to enter the game, and you will try to guess the number that is rolled for each of the three dice (note you submit a guess for each color, e.g. red = 5, green = 2, and blue = 1). If you guess exactly one of the three correctly, then they will give you $1 back. If you guess exactly two out of three correctly, they will give you $5 back. If you guess all three correctly, they will give you $100 back. What is the expected value of this game from your perspective?

6 Divide and Conquer (25 Points)

Suppose we have a two-dimensional array of (positive or negative) integers $A[][]$ such that each each row and column is monotonically increasing. That is, if you consider a row and scan from left to right, the numbers increase. Similarly, if you scan a column from bottom to top, the numbers increase. We are interested in finding the largest number in the array that is less than 0. Give a divide and conquer algorithm which runs in $o(n)$ time (i.e. strictly faster than linear time), where n is the total number of integers in A.