MA/CSSE 473 – Design and Analysis of Algorithms

Homework 1 (29 points total)

These are to be turned in to a drop box on ANGEL. Include your name and the number of the assignment in the filename and also at the top of the first page. If your submission contains multiple documents (for example a Word document and a Maple document), please submit a single ZIP or RAR document that contains all of them. You may write your solutions by hand and scan them if you wish. There is an easy-to-use network scanner in F-217. It will email the scan to you.

The problems to turn in for this assignment are relatively short and simple. But during the first week there is a lot of reading, and there are also many not-for-turnin problems that you should at least think about a little bit.

On many assignments, a couple of the problems to be turned in may require a day or more from the time you first encounter them until the time when the light comes on and you see how to do them. So be sure to read the problem statements right away and start thinking how to do them as soon as you can.

When a problem is given by number, it is from the textbook. 1.1.2 means "problem 2 from section 1.1".

A look ahead to HW 2: Partly because it is due early in the term, I made HW 1 rather short. HW 2 is longer, and is due two days after HW 1. So ideally you should do a few problems from HW 2 before HW 1 is due. But because some students have so many adjustments at the beginning of the term, I am not requiring you to have them done until Wednesday.

Assess your background:

The http://www.rose-hulman.edu/class/csse/csse473/201110/Homework/BackgroundMaterial.html document in the homework folder lists some material from the prerequisite courses and the level of understanding that I hope you will have coming into this course. Different students have had different versions of CSSE230 and MA375, so you should be okay if you are at the suggested level for about 2/3 of the topics. If you are at the suggested level for fewer than half of the topics, you may have quite a bit of remedial work to do. As you can see from the list, most of these topics are reviewed in the 473 textbook. I some cases you will also want to go back to your 230 or 375 textbook for more details. I have also provided some old 230 problems and links to some of my PowerPoint slides from 230 as an additional resource. The problems are at http://www.rose-hulman.edu/class/csse/csse473/201110/Homework/230-problems.pdf

Problems for enlightenment/practice/review (not to turn in, but you should think about them):

How many of them you need to do serious work on depends on you and your background. I do not want to make everyone do one of them for the sake of the (possibly) few who need it. You can hopefully figure out which ones you need to do.

1.1.2	(algorithms patentable?)
1.1.5	(practice Euclid, estimate speedup of Euclid vs. brute force algorithm)
1.1.6,	(prove that the main step of Euclid works)
1.1.7	(Euclid with largest number second)
1.1.8	(smallest, largest number of Euclid divisions ($1 \le m, n \le 10$)
1.1.10a	(Extended Euclid algorithm)
1.2.2	(cabbage, wolf, goat)

1.2.3	(triangle area formulas: which ones are algorithms?)
1.3.1	(ComparisonCountingSort)
1.3.2	(known search algorithms)
1.3.3	(string-matching algorithm)
1.4.1	(efficient delete in an array – this book's "array" behaves more like a Java ArrayList)
1.4.3	(push and pop)
1.4.6	(height of a binary tree)
1.4.7	(inefficient implementations of "priority queue")
1.4.9	(choose best data structure)

Problems to write up and turn in:

1. 1.1.11 (5) (locker doors)
2. 1.2.2 (5) (four people and a flashlight)
3. 1.3.9 (5) (are n given points on circumference of the same circle?). Input: a list of coordinates, output: boolean. You can be brief, but do not be so vague that I cannot tell whether you really know how to do this.
4. 1.4.4 (6) (graph properties, based on adjacency matrix, adjacency list)
5. 1.4.5 (5) Free tree → rooted tree
6. 1.4.10 (3) (anagram checker) Note that it says "anagram", not "palindrome".

A few notes on writing up problems for this course:

- "Produce an algorithm" should be read "produce an efficient algorithm" unless a problem states otherwise.
- A few of the questions in the book can be answered with a word or two. In most cases that is not sufficient; I want to know how you arrived at your conclusion
- You can usually present algorithms as high-level pseudocode (or as real code in a programming language if you wish, though that often takes a lot more time). Go into enough detail to convince me that you are not glossing over any hard parts, but you do not have to give all of the details for the easy parts. Pseudocode, just like real code, often needs comments and explanations. The burden of convincing me that your solution works is on you. I will tell the graders that if they cannot quickly understand your solution, they should give you a small percentage of the points and move on.