MA/CSSE 473 – Design and Analysis of Algorithms

Homework 1

These are to be turned in as hard copy. You can write solutions out by hand, or write them on your computer and print them. If there are multiple pages, please staple them together. Place your name and the number of the assignment at the top of the first page. If you submit multiple pages, please staple them A small stapler that will easily fit in your computer backpack can be purchased for about $2.00)

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 many not-to-turn-in problems that you should think about at least a little bit.

A couple of the problems to be turned in, may require a day or so from the time you first encounter it until the time when the light comes on and you see how to do it. So be sure to start thinking about the problems a couple of days before they are due.

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

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 ($1 \leq m, n \leq 10$)
1.1.10a (Extended Euclid algorithm)
1.2.2 (cabbage, wolf, cabbage)
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 (locker doors)
2. 1.2.2 (four people and a flashlight)
3. 1.3.9 (are all points on circumference of the same circle?)
4. 1.4.4 (graph properties, based on adjacency matrix, adjacency list)
5. 1.4.10 (anagram checker)