Problem Set 1- Due January. 24, 2012 4:10PM

Guidelines for writeups

When describing an algorithm first give a high level overview of your approach, then fill in details as needed to justify your time bound and correctness. Unless you are given a target time bound to achieve, assume you are trying to make your algorithm as fast as possible (in big O terms). Your pseudo-code should always be commented (unless it is self commenting). You are also encouraged to describe your solutions in terms of known algorithms and data structures (e.g. sort the numbers, insert into a balanced binary search tree, delete the minimum of a heap, could all be described with little or no further elaboration).

See the sample writeup (samp.hw.pdf) on the class webpage.

For proofs, strive for clear, clean arguments. It will always be somewhat a matter of taste which steps can be skipped, but try to avoid proofs of obvious points, and be clear on anything tricky. Define your notation carefully. This will often allow you to give a much crisper argument.

Do your own work

It is normal to discuss the homeworks with your classmates at a high level, and particularly to get clarifications of what is being asked (though of course you will get more reliable information from me or Susan). However, the actual detailed solution you turn in is to be your own work. Note that you can probably find solutions on the web, from books, or other students to many of these problems. To turn in such solutions as your own work is dishonest, and may subject you to a campus judicial conduct committee.

Submission details

Homework is due at 4:10 on the due date (it can be turned in to the TA (hard copy or electronic) or put in the homework box in Room 2131 Kemper. Everyone gets two free late days for the quarter. Once you have used up your late days there is a 20% penalty per day for turning in late homework. Also, no homework will be accepted more than two days late.

Problem 1.(20) Problem 6, page 318. Solve this problem as a shortest path problem. Give the time and space requirements for your solution.

Problem 2.(20) Consider a variation on the problem of finding line segments to fit a set of points (as discussed in class and in 6.3). The only change is that we are now given a maximum number of line segments k we can use (we can use fewer than k lines if that gives a better solution, but we can't use more than k). Develop a good solution algorithm to solve this version of the problem. Justify your algorithm's correctness and give its run time.

Problem 3.(25) Shuffle: Suppose you are given three strings of characters: $X = x_1x_2 \dots x_n$, $Y = y_1y_2 \dots y_m$, and $Z = z_1 \dots z_{n+m}$. We want to know whether we can *shuffle* X and Y to produce Z. A shuffle S of two strings X and Y is valid if the characters of X appear **in order** in S and if we remove those characters from S, what remains is Y. For example, if

$X = aabacb$, $Y = caa$, $caaabaacb$ is a shuffle (the first, second and 7th characters give us Y, the rest is X), but $ccaaaabab$ is not (since we can't have two c's appear before any $b's$.

Give an efficient dynamic programming algorithm that given X, Y, Z determines whether Z is a shuffle of X and Y. Analyze your algorithms space and time requirements.

Problem 4.(25) Problem 6.26 on p. 333.

Problem 5.(20) . KT 6.28 on page 334 *Scheduling*