Computer Science 202a

Homework #8, due in class Wednesday, December 6, 2006

Please include the total time you have spent on (1) reading and (2) doing the problems for this assignment. This will help me calibrate the workload; including this information will be worth a *bonus* of 2 points on this problem set. Each problem is worth 10 points; partial credit will be given if the grader can easily understand enough of your answer to award it.

1. A relation on ordered pairs of positive integers. Text 8.5, problem 16, page 563.
2. Hasse diagram for $P(S)$. Text 8.6, problem 24, page 579.
3. Existence of two vertices with the same degree. Text 9.2, problem 18, page 609.
4. Minimum requirement for the number of edges in a connected graph. Text 9.4, problem 26, page 631.
5. Euler circuits in graphs. Please justify your answers. Text 9.5, problem 26, page 645.
6. K_5 on the surface of a bagel. Note that you may represent the surface of the bagel using a rectangle. Text 9.7, problem 36, page 666.
7. Nonisomorphic graphs satisfying certain conditions. Please draw the graphs and argue that they are the only ones satisfying the required conditions. Text, Supplementary Exercises for Chapter 9, problem 24, page 679.
8. Another necessary and sufficient condition for a tree. Text 10.1, problem 14, page 694.
9. A property of binomial trees. Note the definition of binomial trees in the text after problem 12. Text, Supplementary Exercises for Chapter 10, problem 16, page 745.
10. Suppose T is a rooted ordered tree with vertices a, b, c, d, e, f, g, h. The preorder traversal of T gives the list of vertices: $abhcfged$ and the postorder traversal of T gives the list of vertices $hcbfdega$.
    (a) Draw the tree T (with vertices correctly labelled, the root at the top and the children of each node correctly ordered from left to right.)
    (b) Devise a polynomial time algorithm to determine the edges of a rooted ordered tree T given two lists: L_1, the vertices of T in a preorder traversal, and L_2, the vertices of T in a postorder traversal of T. Be sure to explain why you algorithm is correct and why it runs in time bounded by a polynomial in n, the number of vertices in T. Hint: recursion may be helpful.