MAT385 Test 2 (Fall 2007): Sets, Graphs, Trees

Name:

Directions: Problems are **not equally weighted**. Show your work! Answers without justification will likely result in few points. Your written work also allows me the option of giving you partial credit in the event of an incorrect final answer (but good reasoning). Indicate clearly your answer to each problem (e.g., put a box around it). **Good luck!**

Problem 1. (10 points) The preorder and postorder traversals of a tree are given as follows:

$$\begin{array}{ll} preorder: & + * 3\ 6 + 2\ 4 * 9 * 2\ 4 + 6\ 2 \\ postorder: & 3\ 6\ 2\ 4 + * 9\ 2\ 4\ 6\ 2 + * * + \end{array}$$

Hint: the tree I used to do those traversals is not binary.

a. Construct a tree consistent with these traversals (4 pts) (If you're stuck, I'll sell one to you).

b. (1 pt) Write the inorder traversal.

c. (1 pt) Compute the value stored in the expression tree, and give its value here: ___________

d. Since a tree is a graph, you can also do a depth-first and breadth-first traversal of this graph. Start from the root node, and produce both (if you know any shortcuts, use them!;):

   i. (2pts) breadth-first:

   ii. (2pts) depth-first:

Problem 2. (26 points) Consider the graph G, given by its (symmetric) adjacency matrix:

$$G = \begin{bmatrix} 0 & & & & \\ 3 & 0 & & & \\ 6 & 4 & 0 & & \\ \infty & 2 & 5 & 0 & \\ 12 & 8 & 3 & 1 & 0 \end{bmatrix}$$

a. (1 pts) Fill in the rest of the adjacency matrix.

b. (1 pts) Is this a planar graph?

c. (3 pts) Draw the graph in the top right space, using node labels $\{A, B, C, D, E\}$, where A corresponds to the first node of the adjacency matrix, etc. Put the nodes clockwise on a pentagon, with A at the top. If the graph is planar, avoid all arc-intersections; otherwise, minimize their number.

d. (1 pts) Give an adjacency list representation of node A.

e. (4 pts) Bellman-Ford can be written so that it "settles" one node at each iteration as paths of the next longer length are added. The first node settled is the start node, at paths of length 0. We can then eliminate a settled node from further consideration, including all references to the "settled" node from the adjacency lists of the remaining nodes, effectively reducing the number of nodes in the graph by 1. Here's a pseudo-code algorithm:

   i. Let Bellman-Ford choose node x to eliminate; then

   ii. For each node y in the adjacency list of x, do:
   search for and destroy the arc referencing x in y's adjacency list.

Suppose that the algorithm we use for "search and destroy" uses a sequential search as it peruses an adjacency list. Considering a simple graph of n nodes, what is the worst case number of comparisons the algorithm might have to do in carrying out Bellman-Ford completely?

Problem 2, cont.

f. (6 pts) Perform Dykstra's algorithm on the graph G, starting at the node A and going to the node E. Give the initial values of d and s, and then their values after the first two iterations (if necessary – note: the adjacency matrix appears below, with Floyd's):

	A	B	C	D	E
d					
s					

	A	B	C	D	E
d					
s					

	A	B	C	D	E
d					
s					

g. (6 pts) Use Bellman-Ford starting from node A, giving the initial values of d and s and then their values after the first two iterations:

	A	B	C	D	E
d					
s					

	A	B	C	D	E
d					
s					

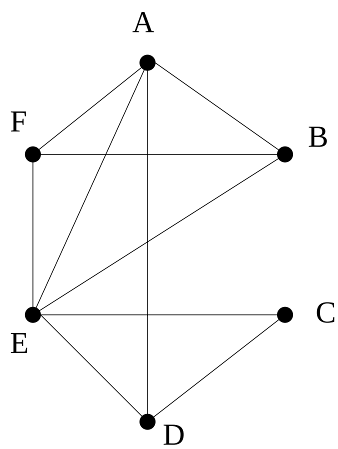
	A	B	C	D	E
d					
s					

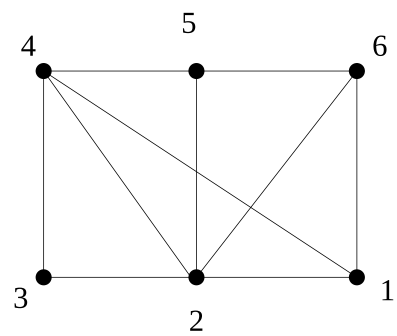
h. (4 pts) Use Floyd's to give the updated adjacency matrix after finishing the $k = 1$ case (checking for shortcuts via A):

$$\text{Initial: } \begin{bmatrix} 0 & & & & \\ 3 & 0 & & & \\ 6 & 4 & 0 & & \\ \infty & 2 & 5 & 0 & \\ 12 & 8 & 3 & 1 & 0 \end{bmatrix} \qquad \text{After k=1: } \begin{bmatrix} 0 & & & & \\ & 0 & & & \\ & & 0 & & \\ & & & 0 & \\ & & & & 0 \end{bmatrix}$$

Problem 3. (14 points total)

a. (6 pts): Demonstrate whether the two graphs are isomorphic or non-isomorphic





b. Show all possible distinct ways to inter-connect identical computers (with single cables – not connecting two is an acceptible choice, too). There is one way to connect one computer: it is connected to nothing else. So we don't connect computers to themselves.

i. (2pts) two computers:

ii. (2pts) three computers:

iii. (4pts) four computers:

c. **Extra credit** (2pts): Can you figure out a rule for the number of isomorphic simple graphs with n nodes?

Problem 4. (10 points) Consider G, a weighted, connected graph of n nodes.

a. (2 pts) How many arcs will be in a minimal spanning tree of G?

b. (2 pts) What is the minimum depth possible for a minimal spanning tree of G? What is the maximum possible depth?

c. (2 pts) What is the minimum possible depth for a binary search tree of n nodes?

d. (2 pts) Given the distinct letters in the words "the distinct letters", find an order that achieves the minimal depth for the binary search tree, and draw the tree.

e. (2 pts) If your binary search tree above were a filter to pick up just those letters from ordinary English text, and you wanted to minimize the number of comparisons (given random text), would you change your tree above? What other considerations become important?

Problem 5. (10 points) True or False? (Fix any false statements.)

a. ( ) A Hamiltonian Circuit exists for any complete graph with at least 3 nodes.

b. ( ) If there is an Euler path on a graph, the graph can be drawn without lifting one's pencil from the paper.

c. ( ) The power set of the empty set is empty.

d. ( ) The integers and the real numbers are the same size as sets.

e. ( ) Binary search is optimal in its worst case behavior.