CS 4102 Exam 3

Name

You MUST write your e-mail ID on **EACH** page and put your name on the top of this page, too.

If you are still writing when "pens down" is called, your exam will be ripped up and not graded – sorry to have to be strict on this!

There are 4 pages to this exam. Once the exam starts, please make sure you have all the pages. Questions are worth different amounts of points.

Answers for the short-answer questions should not exceed about 20 words; if your answer is too long (say, more than 30 words), you will get a zero for that question!

This exam is CLOSED text book, closed-notes, closed-calculator, closed-cell phone, closed-computer, closed-neighbor, etc. Questions are worth different amounts, so be sure to look over all the questions and plan your time accordingly. Please sign the honor pledge below.

The Tao that is seen
Is not the true Tao,
until You bring fresh toner.

Page 2: Assorted Questions on Many Topics

1. [8 points] In using a decision tree proof to find a lower bound for sorting by comparison of keys,

   (A) What do the leaves of the decision tree represent? ________________

   (B) How many leaves must there be? ________________

   (C) What do the internal nodes represent? ________________

   (D) Give the formula for the minimum height of the decision tree in terms of the number of leaves. (The formula, not the order-class.)

2. [10 points] For each of the following statements that are true, circle the letter identifying the statement. (Zero or all of them could be true.)

   (A) When visiting a node v during DFS on an <u>undirected</u> graph, each node w that is adjacent to v and is colored gray indicates the graph has a cycle.

   (B) For an undirected graph G, it is not possible for the BFS tree to have greater height than G's DFS tree.

   (C) The time-complexity for using DFS to count connected components in an undirected graph is $\Theta(V(V+E))$

   (D) When visiting a node during DFS on an <u>directed</u> graph, neither cross edges or forward edges indicate a cycle.

   (E) BFS can be used to tell if an undirected graph G is bipartite by checking if a non-tree edge connects nodes on the same level.

3. [3 points] Say the directed edge from u to v in a circulation graph has a lower bound x on its capacity. To convert this to a circulation graph <u>without</u> lower bounds on the edges, what are the new values of d_u and d_v (where d_i is the demand associated with vertex i)?

   New $d_u =$: ______________________ New $d_v =$: ______________________

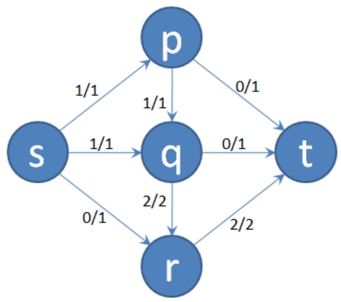
4. [3 points] Explain what it means for a node v in a circulation graph (without lower bounds) to be in equilibrium in terms of the flows on its edges and its demand d_v.

Page 3: Getting into the flow of things

5. [6 points] Prove the following statement is true. You may write your argument in natural language, but your argument should still be logically formal:

   If a flow network has no augmenting path with respect to the current flow f, then there exists a cut C whose capacity is exactly f.

Take a look at the flow network given below:



6. [6 points] Given the graph G above, draw the residual graph G'.

7. [6 points] Find an augmenting path in the graph. List the nodes in the path you found in order (e.g., s-a-b-c-d-t)

8. [6 points] Assume an algorithm (e.g., Ford-Fulkerson) has pushed as much flow as possible through the augmenting path that you found. Draw the updated residual graph G' before the next iteration of the algorithm begins.

Page 4: Applying Algorithms to Solve Problems

9. [6 points] Given an undirected, unweighted graph G, the *eccentricity* of a node v is the largest of the shortest possible distances from v to any other node in G, where distance is number of edge traversals. The *minimum eccentricity* in G is called the *radius* of G.

   Making use of at least one algorithm studied in this unit, describe the steps you would take to find the radius of an input graph G. (You might need to modify an algorithm we've studied to return a certain value.)

10. [6 points] Give the order-class for the worst-case time-complexity of your algorithm.

11. [6 points] You've been given two empty water jugs (i.e. containers). One holds exactly 5 gallons, and the other holds exactly 3 gallons. You must find a sequence of "steps" that puts exactly 4 gallons into the larger jug. At each step, you can either fill a jug from a fountain of water, empty a jug, or by pour water from one jug to the other.
    (BTW, in the movie *Die Hard with a Vengeance*, the Bruce Willis character solved this in five minutes to stop a bomb from going off. Surely you can do just as well! Yippee-ki-yay!)

    What to do: Use state-space search to show how you would *start* to solve this problem. But do not draw the entire search-tree (there's probably not enough time). You just need to draw the tree's first 6 nodes (including the start state). Be sure we can understand what information each state stores.