Computer Science 789 Spring 2009 Final Examination, May 5, 2009

Name:_____

No books, notes, scratch paper, or calculators. Use pen or pencil, any color. Use the rest of this page and the backs of the pages for scratch paper. If you need more scratch paper, it will be provided.

The entire examination is 180 points.

1. True or False. [5 points each]

   (a) _____ Every finite metric space embeds in $\mathbb{R}^n$ with the $\mathcal{L}^1$ metric, for some n.

   (b) _____ Every finite metric space embeds in $\mathbb{R}^n$ with the $\mathcal{L}^2$ metric, for some n.

   (c) _____ Every finite metric space embeds in $\mathbb{R}^n$ with the $\mathcal{L}^\infty$ metric, for some n.

2. [5 points] What is the best known upper bound on the competitiveness for the 3-server problem in a general metric space? Who proved it? (Two names.)

3. [5 points] What is the best known lower bound on the competitiveness for the 3-server problem in a general metric space? Who proved it? (Two names.)

4. [5 points] What is the least competitiveness of any randomized trackless algorithm for the k-paging problem, for general k? What is that algorithm? (Just the name of the algorithm will do.)

5. [5 points] The work function algorithm is known to be _____-competitive for any metrical task system with n states.

6. [5 points] Draw a picture of the *tight span* of the 4-point metric space M whose distance function is given by the matrix below.

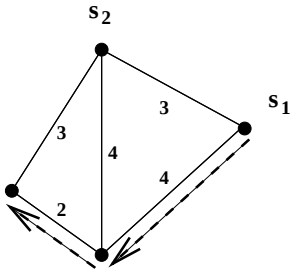
	x	y	z	w
x	0	4	4	4
y	4	0	4	4
z	4	4	0	6
w	4	4	6	0

7. [20 points] Consider the 2-server problem in $\mathcal{M}_5$, the class of all metric spaces where distances are integers not more than five.

   (a) Suppose that, in the initial configuration, our servers, which we name s_1 and s_2, are 4 apart. Assuming that the optimal servers match ours initially, draw a figure representing the initial trackless work function.

   (b) The first request is to a point which is 4 from s_1 and 4 from s_2. We serve the request by moving s_1. Draw a figure representing the resulting trackless offset function.

   (c) The second request is to a point that is 2 from s_1 and 3 from s_2. We serve the request by moving s_1 again. Draw a figure representing the resulting trackless offset function.



8.

[20 points] Consider the following metrical task system. There are two states, which we call A and B, and the cost of moving from one to the other is 2. There are two tasks, which we call α and β. The cost matrix is as given.

	A	B
α	0	1
β	2	0

(a) Draw an appropriate red-black game graph.

(b) Compute the competitiveness of this metrical task system. Indicate potentials and critical edges in the red-black game graph.

9. [20 points]There are two standard models for describing randomized online algorithm, *behavioral* and *distributional*. In addition, there is one other model which is used to describe *barely random* algorithms. (Although some might argue that you do not actually need a third model, but rather, that barely random algorithm can be described using one of the two standard models.)

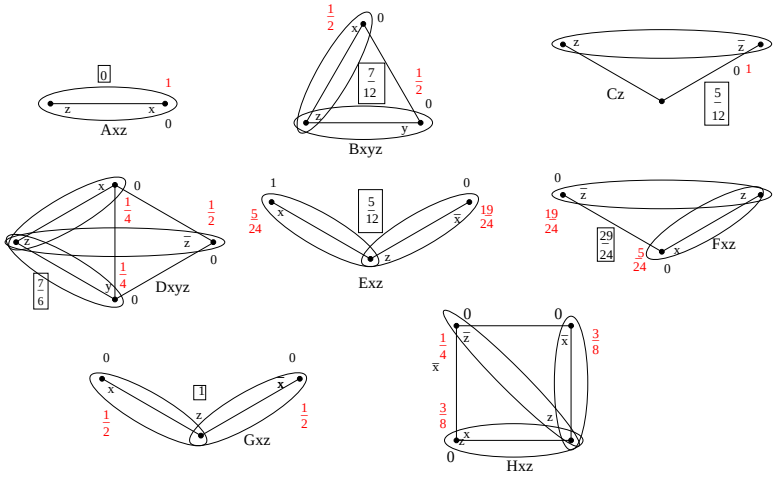
   (a) Give the definition of a randomized online algorithm using the behavioral model.

   (b) Give the definition of a randomized online algorithm using the distributional model.

   (c) Give the definition of a barely random online algorithm.

10. [20 points]Prove that the competitiveness of HARMONIC for the k-server problem cannot be less than $\binom{k+1}{2}$. (Hint: What is the *lazy adversary*?)

11. [20 points] (A problem based on Exercise 11.2.) Consider a knowledge state algorithm $\mathcal{A}$ whose knowledge states are as illustrated in the figure below. Give a correct move of $\mathcal{A}$ if the knowledge state is Cx and the request is some point $r \neq x$.



12. [20 points] (A problem based on Exercise 12.2.) Suppose you are given a red-black game graph $\mathcal{G}$, where the set of red nodes is $\mathcal{R}$ and the set of black nodes is $\mathcal{B}$, and you need to write a computer program for finding the competitiveness of the game, as well as finding an optimal strategy for the red player. You are also given that

    (a) Every black edge has a non-negative weight.

    (b) Every red edge has a non-negative weight.

    (c) No node is a *sink*, *i.e.*every node has at least one out-edge. (Thus, the game will continue forever as long as the black player continues to make moves.)

    Describe how you would design your program. I don't mean for you to write pseudo-code: everything can be explained using English sentences, with a few mathematical expressions thrown in. (If you feel like you absolutely **must** write pseudo-code, go ahead, but don't make it too complicated. I might not be in "compiler" mode when I grade this.) Partial credit on this problem is possible. If you describe a program that simply finds an optimal strategy for the red player, given that the competitiveness is known, then that's worth partial credit.

    Do not worry about the time complexity of your program.

13. [20 points](Another problem based on the homework.)