Computer Science 789 Spring 2009 Examination, March 31, 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 120 points.

1. [5 points] Give the definition of a metric space.

2. [5 points] Give the definition of a metrical task system.

3. [5 points] Give the definition of the competitiveness of a deterministic online algorithm for a metrical task system.

4. [5 points] Give the definition of the competitiveness of a randomized online algorithm for a metrical task system against the oblivious adversary.

5. [20 points] Prove that *Move-to-Front* is 2-competitive for the list accessing problem.

6. [20 points] Given a metric space M that has at least $k+1$ points, prove that there is no online algorithm for the k-server problem in M whose competitiveness is less than k.

7. [20 points] Consider the 2-server problem on a metric space M consisting of three points, a, b, and c, where $d(a,b) = \varepsilon$, $d(b,c) = 1$, and $d(a,c) = 1 + \varepsilon$, where $0 < \varepsilon < 1$. The initial position of the servers is $\{a, c\}$. The request sequence is $(ba)^N$, where N is a large integer.

   (a) M embeds in the line. Thus, you can use the virtual double cover algorithm for this problem. Show the positions of the two (virtual) servers in the line at each step. How many requests does it take for the two servers to reach the configuration $\{a, b\}$?

   (b) Show the behavior of the work function algorithm, computing the work function (or the offset function, if you prefer) at each step. How many requests does it take for the two servers to reach the configuration $\{a, b\}$?

   Hint: First try to work the problem for a specific value, such as $\varepsilon = \frac{3}{5}$. Then, try a smaller value. The smaller ε, the more requests it takes.

8. [20 points] The expected cost of a move of EQUITABLE is usually defined as the transportation cost of changing a distribution. But if want a behavioral implementation of EQUITABLE, the moves must to be defined taking into account the actual cache, not just the probability that we have a certain cache.

Consider the 2-paging problem, where there are pages a, b, and c, and where the offset function is $a|bc|$. What is the cost if the request is for page c? We would say, $\frac{1}{2}$. But really, it's either 0 or 1, depending on whether our cache is actually $\{a, b\}$ or $\{a, c\}$. If the request is to a new page r, EQUITABLE pays 1, but how does it choose which page to eject?

The following pseudocode, written in a Pascal-like language whose meaning should be clear, is a behavioral implementation of EQUITABLE in the case that the offset function is $a|bc|$ and our cache is $\{a, b\}$.

```
procedure moveInBarInOutBar(a,b,c,r:page);
  {input conditions: offsetfunction = a|bc|, cache = a,b, request = r}
 begin
  if r = a then { do nothing }
  else if r = b then { this is a hit, but we still update the offset function }
   offsetfunction := ab||
  else if r = c then
   begin
    offsetfunction := ac|| ;
    eject b
   end
  else { r is a new page }
   begin
    offsetfunction := r|abc| ;
    rand
     2/3: eject a;
     1/3: eject b
    end { end of randomization construction }
   end { end of case r = new page }
 end; { end of procedure moveInBarInOutBar }
```

Now, suppose that $k = 3$, the offset function is $a|bc|de|$, and the next request is a page r, which could be one the known five pages or a new page. Write, in the style above, the two procedures you need to encode a move of EQUITABLE in this situation.

Hint: This problem is requires a substantial amount of calculation. The numbers can get more complex than you might think: at one point, you will eject a page with probability $\frac{14}{25}$, for example.

You should be able write your work on two pages; three if your writing is large. (That's not counting the pseudo-code.)

All but six of you got the correct distribution of EQUITABLE for this offset function in the homework. Those six people should take a careful look at their previous work.

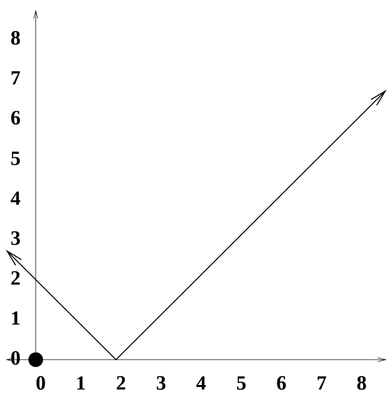
9. [20 points]

For the 2-server problem on the real line, work functions (and offset functions) can be represented as continuous piecewise linear curves, where each piece has slope plus or minus 1.

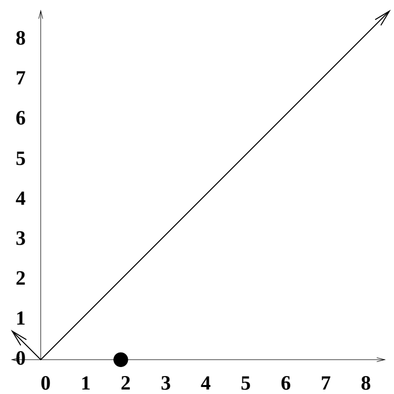
Take the start configuration to be $\{0, 2\}$, where 0 is designated to be the last request point. The figures below show the sequence of offset functions, given the request sequence $2, 6, 2$. In each figure, if r is the last request point, there will be a large dot at r, and the function of x shown is $\omega(r, x)$. Thus, in the first figure, the curve $\omega^0(0, x)$ is shown, in the second figure $\omega^1(2, x)$, in the third figure $\omega^2(6, x)$, and in the fourth figure $\omega^3(2, x)$, where ω^i is the offset function after i requests.

For example, in the initial configuration, $\omega(0, 2) = 0$, $\omega(0, 1) = \omega(0, 3) = 1$, $\omega(0, 4) = 2$, $\omega(0, 20) = 18$ (off the chart, of course), *etc.*.

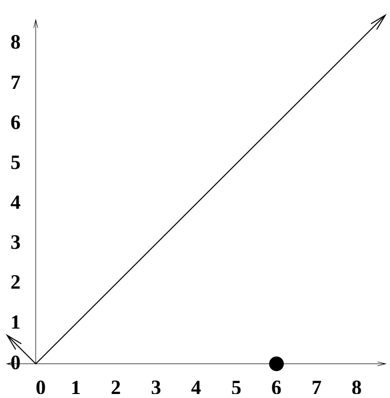
We are representing a function in two variables by a function in one variable. In fact, the figure is actually a 2-dimensional slice of a 3-dimensional graph, namely the graph of the offset function on two variables. But we don't really lose any information, since, if a pair $\{a, b\}$ is in the support of the work function (or offset function), then either a or b must be the last request point, and hence that slice contains the portion of the graph over the support.



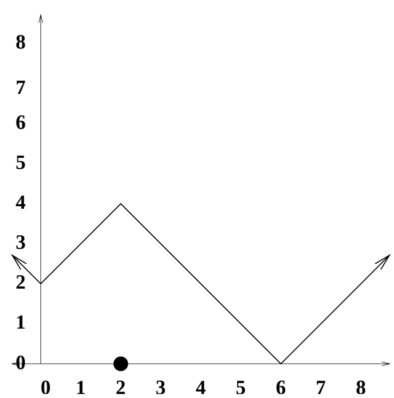
Initial Offset Function: $\omega^0(0, x)$



After Request 2: $\omega^1(2, x)$



After Requests 2,6: $\omega^2(6, x)$



After Requests 2,6,2: $\omega^3(2, x)$

Note that $\omega^1 = \omega^0$, but the figure looks different because the last request is 2 instead of 0. (The two figures show different slices of the same graph.)

(a) Compute the amortized optimal cost of each of the three steps. (Hint: the first one is zero.)

(b) Suppose the fourth request is to the point 3. Draw the offset function after the request sequence 2,6,2,3. What is the amortized optimal cost of the fourth step?