CS 369 - Problem Set One

Due Date: Thursday, May 2, 2013, by end of lecture

- Please make your answers clear and concise. In most cases, you should not need more than the equivalent of a page with size 12 font typed. You're more likely to get points subtracted if your proof is difficult to read or understand. To improve legibility, type your solutions if you can.
- Each solution should start on a new page.
- Searching for answers on the Web is not allowed.

1. **(10 points)** Consider the least frequently used (LFU) algorithm. It keeps a counter for the number of accesses to each page. When a cache miss occurs, it evicts the page with the smallest value of the counter (ties are broken arbitrary). Show that LFU has an unbounded competitive ratio. In other words, for every R, give a sequence such that the number of misses due to LFU is at least R times more than the number of misses due to the best offline policy.

2. **(10 points)** Consider a randomized online paging algorithm that, on each fault, evicts a (uniformly) random page from the cache. Prove that its competitive ratio is $\Omega(k)$, where k is the number of pages that fit into the cache. [Hint: focus on a simple case where cache size is k and there are only $k+1$ different pages.]

3. **(15 points)** Show how to extend the $1/\infty$ case studied in class to $p(k)/\infty$ case, i.e. job k adds load $p(k)$ and can be assigned to a subset of machines $m(k)$. Note that you will get the full credit even for $2\log(n)$-competitive analysis. [Go through the proof presented in class and show what changes are necessary, both in definitions and in claims.]

4. **(20 points)**

   - **(10pt)** prove that the greedy algorithm for $1/\infty$ case presented in class is $\log(n)$ competitive for the case where the load on each machine is defined by $\sum_j p_i(j) \cdot Q_i$ where sum is over the jobs assigned to the machine and $p_i(j)$ is the load of job j on machine i which is either 1 or ∞. Also Q_i is an arbitrary integer scaling factor for machine i.
   - **(10pt)** Use the previous part and extend the approach we used for $1/\infty$ case to design an algorithm for unrelated machines that achieves $O(\log n \log Q)$ competitive ratio, where you are told that $1 \leq p_i(j) \leq Q$ for some given Q. [Hint: look at each machine as a logarithmic number of machines of exponentially increasing speeds.]

5. **(20 points)** Consider load balancing on unrelated machines. Prove that competitive ratio of greedy algorithm (i.e. the algorithm which at each step assign the current job to the machine which execute it with lowest resulting load) is bounded by $\Omega(n)$.