Advanced Algorithms – Exercise 4

Deadline: 2/2/2006

1. The goal of this question is to analyze the competitive ratio of the following naive randomized algorithm for the paging problem: whenever a page fault occurs, choose a page to evict from the cache uniformly at random. We denote by k the number of pages in the cache.

   (a) In class we saw how to prove upper bounds for the competitiveness of online deterministic algorithms using the method of potential functions. Slightly modify this method to enable us to prove upper bounds on the expected cost of randomized algorithms. Prove correctness.

   Hint: For proving correctness use induction and conditional expectations.

   (b) Let ALG_t denote the set of pages in the cache of the naive randomized algorithm described earlier at stage t. Let ADV_t denote the set of pages in the cache of the adversary. For every stage t, let $\phi_t = k \cdot |ADV_t - ALG_t|$ be the potential function (meaning: the number of pages which are in the adversary's cache but not in the algorithm's cache.) Prove that the naive randomized algorithm is only k-competitive.

2. In this question we will prove a lower bound for the competitiveness of any deterministic online algorithm for the online scheduling problem. Recall that in this problem we are given n machines, and jobs (with different weights) come in an online manner. Our goal is to minimize the makespan. In class we saw a $2 - \frac{1}{n}$ upper bound for this problem.

   (a) Prove that this bound is tight for the case where $n = 2$.

   (b) Prove that this bound is tight for the case where $n = 3$.

   Hint: Consider the following sequence of jobs' weights: 1,1,1,3,3,3,6.

3. This question characterizes the deterministic communication complexity of the COMMON-1-BIT problem. An instance of the COMMON-1-BIT problem is a pair $(x, y) \subseteq \{0,1\}^n \times \{0,1\}^n$, which we interrupt in the following way: Alice holds an n bit string $x = x_1 x_2 ... x_n$, and Bob holds an n bit string $y = y_1 y_2 ... y_n$. The goal of the COMMON-1-BIT problem is to determine if there exists some index i such that $x_1 = y_i = 1$.

   (a) Show a (trivial) upper bound for the deterministic communication complexity of the COMMON-1-BIT problem (You need to show a protocol that solves this problem. The number of bits needed for this protocol in the worst case is an upper bound for the communication complexity of the problem.)

   (b) Show that there exists a *fooling set* of size 2^n. That is, you need to prove that there exists a set $F \subseteq \{0,1\}^n \times \{0,1\}^n$, $|F| = 2^n$, where for every $I_1, I_2 \in F$, $I_1 \neq I_2$, the sequence of bits generated for the instance I_1 must be different than the sequence of bits generated for the instance I_2. You may want to prove the following lemma:

Lemma: Let $(x_1, y_1), (x_2, y_2) \in \{0,1\}^n \times \{0,1\}^n$, $(x_1, y_1) \neq (x_2, y_2)$. Suppose there is a protocol that generates the same sequence of bits for both instances (x_1, y_1) and (x_2, y_2). Then the same sequence of bits must also be generated by the protocol for (x_1, y_2) and for (x_2, y_1).

(c) Deduce that the any deterministic protocol that solves the COMMON-1-BIT uses at least n bits.

Remark: It is actually quite easy to see that there is a fooling set of $2^n + 1$ (can you find it?) Therefore, the communication complexity of the COMMON-1-BIT problem is exactly $n + 1$.

4. Consider the following hat problem: there are n players, each wears exactly one hat, which is either red or blue. A player can not see his own hat, but can see all other hats. No communication between of the players is allowed. Each player has to guess the color of the hat he is wearing.

   (a) Describe a randomized algorithm such that half of the players which wear a red hat guess (in expectation) red, and half of the players which wear a blue hat guess (in expectation) blue.

   (b) Let $c \in \{r, b\}$ be the input vector (that is, c_j denotes if the j'th player hat is red or blue.) For every player i define the following notation: let $I_r^i = \Sigma_{j|j \neq i, c_j = r} j$. I.e., the sum of indices of the players that have red hats, *not including* player i. Let $B_r^i = |\{j | j < i, c_j = r\}|$: the number of players with red hats before player i. Let $M_r^i = i + I_r^i + B_r^i$. Notice that the value of M_r^i does not depend on c_i.

   The following is a *deterministic* algorithm in which half (rounded down) of the players which wear a red hat guess red, and half (rounded down) of the players which wear a blue hat guess blue: For each player i, if $M_r^i \equiv 0 \pmod 2$ then guess that c_i is red, otherwise guess it is blue.

      i. Prove that half (rounded down) of the players that wear a red hat will guess red. **Hint:** Prove that every other player that wears a red hat guesses red.

      ii. Define M_b^i in an analogous way to M_r^i, with respect to blue hats. Prove that $M_b^i + M_r^i$ is a constant (modulo 2).

      iii. Prove that half (rounded down) of the players that wear a red hat will guess blue. **Hint:** Use the fact that $M_b^i + M_r^i$ is a constant (modulo 2), and use a similar argument to the one in the first bullet.

   (c) Consider a simplified version of the digital goods auction we saw in class: we are interested in selling one good for each on of the n bidders. Each bidder i has a private value v_i. In the simplified version each bidder's value may take only two possible values: a high value h, or a low value l. A bidder will buy the item if the price p_i he is offered is smaller than v_i. Our goal is to maximize the revenue. As in class, you may assume that at least 2 players have a private value of $\max_i v_i$.

   Show a deterministic $O(1)$-competitive incentive-compatible algorithm for this problem.