15896 Spring 2015 Homework #3:
Matching and Noncooperative Games

Ariel Procaccia (arielpro@cs.cmu.edu)

Due: 4/7/2015 1:30pm

Rules:

- You may work with a friend on the problems if you want. If you do work with a friend, please submit a single solution.
- If you've seen a problem before (some of them are "famous"), then say that in your solution. Also, if you use any sources other than the AGT book, write that down too. **Please don't deliberately search for solutions online!** It's fine to look up a complicated sum or inequality or whatever, but don't look up an entire solution.
- Please prepare a pdf with your solution and send it to me by email.

Problems:

1. [**20 points**] In the Cycle Cover problem, we are given a directed graph and two integers k, t; we are asked whether it is possible to cover at least t vertices with disjoint cycles of length at most k. We stated in class (lecture 15, slide 11) that the Cycle Cover problem is NP-hard when there is a given upper bound k on the length of cycles. Show that if there is no such upper bound then the problem can be solved in polynomial time.

   Hint: You may rely on the fact that a maximum weight perfect matching in a bipartite graph can be computed in polynomial time.

2. [**25 points**] We proved in class (lecture 16, slides 5–7) that when there are at least two players, no deterministic strategyproof kidney exchange mechanism can provide an α-approximation for $\alpha < 2$. Show that no *randomized* strategyproof kidney exchange mechanism can provide an α-approximation for $\alpha < 6/5$.

   Hint: You can use exactly the same examples we used in class to establish the deterministic case.

3. Consider the following scheduling game. The players $N = \{1, \ldots, n\}$ are associated with tasks, each with weight w_i. There is also a set M of m machines. Each player chooses a machine to place his task on, that is, the strategy space of each agent is M. A strategy profile induces an assignment $A : N \to M$ of agents (or tasks) to machines; the *cost* of player i is the total load on the machine to which i is assigned — $\ell_{A(i)} = \sum_{j \in N:\ A(j)=A(i)} w_j$. Our objective function is the *makespan*, which is the maximum load on any machine: $\text{cost}(A) = \max_{\mu \in M} \ell_\mu$. It is known that scheduling games always have pure Nash equilibria.

(a) **[35 points]** Let G be a scheduling game with n tasks of weight $w_1, \ldots, w_n$, and m machines. Let $A : N \to M$ be a Nash equilibrium assignment. Then

$$\mathrm{cost}(A) \leq \left(2 - \frac{2}{m+1}\right) \cdot \mathrm{opt}(G).$$

That is, the (pure) price of anarchy is at most $2 - 2/(m+1)$.

(b) **[20 points]** Prove that the upper bound of part (a) is tight, by constructing an appropriate family of scheduling games for each $m \in \mathbb{N}$.