Economics and Computation (Spring 2026)
Assignment #4

Due: 4/14/2026 11:59pm ET

Problem 1: Indivisible goods

[**25 points**] Recall that envy-freeness up to *any* good (EFX) is known to be feasible for up to three players with additive valuations, but it is an open problem if it is feasible for four or more players.

Assume, then, that the n players have additive, *identical* valuations, i.e., for all $g \in G$ and $i, j \in N$, $V_i(g) = V_j(g)$. Under this assumption, design a polynomial-time algorithm that computes an EFX allocation.

Problem 2: Online matching algorithms

[**25 points**] Consider a variant on the worst-case online matching model introduced in Slide 7 of Lecture 13. As before, an adversary constructs the set of vertices that will arrive, but the arrival order of these vertices is uniform at random. If the algorithm is deterministic, the "game" proceeds as follows: we announce the algorithm, the adversary constructs the n vertices in V (i.e., defines the edges incident on each vertex in V), and then a random permutation π of V determines the arrival order. The algorithm has competitive ratio α if $\mathbb{E}[ALG(G, \pi)]/OPT(G) \geq \alpha$ for every graph G, where the expectation is taken over the randomness of the permutation π.

Prove that, in this model, the competitive ratio of a deterministic algorithm must be at most 3/4.

Problem 3: Kidney exchange

[**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 14, slide 6) 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.

Problem 4: Stable matching

[**30 points**] Prove that no bipartite matching mechanism with two-sided preferences is strategyproof (on both sides) and stable.

Guidance: Consider an instance with three students, three courses, and the following preferences:

s_1	s_2	s_3
t_1	t_2	t_1
t_2	t_1	t_2
t_3	t_3	t_3

t_1	t_2	t_3
s_2	s_1	s_1
s_1	s_2	s_2
s_3	s_3	s_3

There are two stable matchings; argue that in each them, one of the players can report a ranking leading to a unique stable matching that is (truly) better for them.