# CS261: Exercise Set #8

For the week of May 18–22, 2015

**Instructions:**

(1) *Do not turn anything in.*

(2) The course staff is happy to discuss the solutions of these exercises with you in office hours or on Piazza.

(3) While these exercises are certainly not trivial, you should be able to complete them on your own (perhaps after consulting with the course staff or a friend for hints).

## Exercise 36

Consider the following LP relaxation of the maximum (unweighted) matching problem in general (not necessarily bipartite) graphs:

$$\max \sum_{e \in E} x_e$$

subject to:

$$\sum_{e \in \delta(v)} x_e \leq 1 \qquad \text{for all } v \in V$$

$$x_e \geq 0 \qquad \text{for all } e \in E.$$

Prove that there exists a graph for which the integrality gap of this relaxation is at most $\frac{2}{3}$.

## Exercise 37

Consider an optimization problem $\Pi$ and an LP relaxation for it — formally, a mapping $h$ from instances of $\Pi$ to linear programs, such that every feasible solution to an instance $I \in \Pi$ can be mapped to a feasible solution to the linear program $h(I)$ that has equal objective function value.

Suppose that:

1. The worst-case (over $I$) integrality gap of the LP relaxation is $\alpha$.

2. There is an LP-based $\alpha$-approximation algorithm $\mathcal{A}$. That is, for every $I \in \Pi$, $\mathcal{A}$ outputs a solution with objective function value within an $\alpha$ factor of that of the optimal solution to the LP relaxation $h(I)$.

Prove that for every instance $I \in \Pi$ that realizes the worst-case integrality gap — i.e., the ratio between the objective function values of the optimal solution to $I$ and the optimal solution to $h(I)$ is precisely $\alpha$ — the algorithm $\mathcal{A}$ outputs an optimal solution to the instance $I$.

## Exercise 38

Recall from Lecture #15 that strongly $NP$-hard problems do not admit pseudopolynomial-time algorithms (unless $P = NP$).[1] The point of this exercise is to show that strong $NP$-hardness typically also rules out an FPTAS.[2]

We illustrate this point using the Knapsack problem.[3] Suppose you are given an FPTAS $\mathcal{A}$ for the Knapsack problem. Prove that $\mathcal{A}$, with a suitable choice of the parameter $\epsilon$, is a pseudopolynomial-time (exact) algorithm for the Knapsack problem.[4]

## Exercise 39

Suppose we generalize the online bipartite matching problem studied in lecture by allowing the edges to have arbitrary nonnegative weights. (When a right-hand side vertex shows up, the online algorithm learns its incident edges and their weights.) Prove that for every $\alpha > 0$, there is no deterministic $\alpha$-competitive online algorithm for this problem.

## Exercise 40

Consider the following online matching problem in general, not necessarily bipartite graphs. No information about the graph $G = (V, E)$ is given up front. Vertices arrive one-by-one. When a vertex $v \in V$ arrives, and $S \subseteq V$ are the vertices that arrived previously, the algorithm learns about all of the edges between $v$ and vertices in $S$. Equivalently, after $i$ time steps, the algorithm knows the graph $G[S_i]$ induced by the set $S_i$ of the first $i$ vertices.

Give a $\frac{1}{2}$-competitive online algorithm for this problem.

---

[1] Recall that a *pseudopolynomial-time algorithm* runs in polynomial time for the special case of instances with input numbers that are polynomially-bounded integers.

[2] Recall from Lecture #15 that a *fully polynomial-time approximation scheme (FPTAS)* for a maximization problem takes as input a problem instance and a parameter $\epsilon$, and returns a feasible solution with objective function value at least $(1 - \epsilon)$ times the maximum possible, in time polynomial in the size of the instance and in $\frac{1}{\epsilon}$.

[3] The input is $n$ items with values $v_1, \ldots, v_n$ and weights $w_1, \ldots, w_n$, and a knapsack capacity $W$. Assume that all of these numbers are positive integers. The goal is to choose $S \subseteq \{1, 2, \ldots, n\}$ to maximize $\sum_{i \in S} v_i$ subject to $\sum_{i \in S} w_i \leq W$.

[4] Since the existence of an FPTAS implies the existence of a pseudopolynomial-time exact algorithm, impossibility results for the latter (i.e., strong $NP$-hardness) translate immediately to the former.