Problem Set 8

This problem set is due **at 11:59pm** on **Friday, April 24, 2015.**

Each submitted solution should start with your name, the course number, the problem number, your recitation section, the date, and the names of any students with whom you collaborated.

Exercise 8-1. Read CLRS, Chapter 29.

Exercise 8-2. Exercise 29.2-2.

Exercise 8-3. Exercise 29.2-4.

Exercise 8-4. Read CLRS, Chapter 34.

Exercise 8-5. Exercise 34.2-8.

Exercise 8-6. Exercise 34.3-5.

Problem 8-1. A Simple Simplex Example [25 points] Consider a linear program (LP) consisting of two variables x_1 and x_2 satisfying the following three constraints:

$$\begin{aligned} x_1 + x_2 &\leq 10 \\ x_2 &\geq 4x_1 - 20 \\ x_1 + 3x_2 &\leq 24 \\ x_1, x_2 &\geq 0 \end{aligned}$$

The goal is to maximize the value of the objective function $p = 4x_1 + x_2$.

(a) [5 points] Draw a diagram of the feasible region.

(b) [5 points] Write the given LP in standard form, and transform this standard form representation into slack form.

(c) [10 points] Use Simplex to solve the resulting slack form LP. Identify the pivots you choose and give the resulting modified LPs and the successive feasible solutions. Indicate the successive solutions on your diagram from Part (a).

(d) [5 points] Give the dual LP of your standard-form LP from Part (b) and give its optimal value. (*Hint: Use your solution to Part (c).*)

Problem 8-2. NP-Completeness [25 points]

In this problem, you will prove NP-completeness of a few decision problems. To prove NP-hardness, you may reduce from any problem that has been shown, in class or in CLRS, to be NP-complete.

(a) [5 points]

Let TRIPLE-SAT denote the following decision problem: given a Boolean formula ϕ, decide whether ϕ has at least three distinct satisfying assignments. Prove that TRIPLE-SAT is NP-complete.

(b) [10 points] In Problem Set 1, we considered how one might locate donut shops at some of the vertices of a street network, modeled as an arbitrary undirected graph $G = (V, E)$. Each vertex u has a nonnegative integer value $p(u)$, which describes the potential profit obtainable from a shop located at u. Two shops cannot be located at adjacent vertices. The problem was to design an algorithm that outputs a subset $U \subseteq V$ that maximizes the total profit $\sum_{u \in U} p(u)$. No doubt, you found an algorithm with time complexity that was exponential in the graph parameters. Now we will see why.

Define DONUT to be the following decision problem: given an undirected graph $G = (V, E)$, given a mapping p from vertices $u \in V$ to nonnegative integer profits $p(u)$, and given a nonnegative integer k, decide whether there is a subset $U \subseteq V$ such that no two vertices in U are neighbors in G, and such that $\sum_{u \in U} p(u) \geq k$. Prove that DONUT is NP-hard. (*Hint:* Try a reduction from 3SAT.)

Also, explain why this implies that, if there is a polynomial-time algorithm to solve the original problem, i.e., to output a subset U that maximizes the total profit, then P = NP.

(c) [10 points] Suppose we have one machine and a set of n tasks $a_1, a_2, \ldots, a_n$. Each task a_j requires t_j units of time on the machine, yields a profit of p_j, and has a deadline d_j. Here, the t_j, p_j, and d_j values are nonnegative integers. The machine can process only one task at a time. Not all tasks have to be run, but if a task starts running, it must run without interruption and must complete by its deadline.

A *schedule* for a subset of the tasks describes when each of the tasks in the subset starts running. A schedule must observe the constraints given above. The *profit* for the schedule is the sum of all the p_j values for the tasks a_j in the schedule.

The problem is to produce a schedule for a subset of the tasks that returns the greatest possible amount of profit. State this problem as a decision problem and show that it is NP-complete. In showing this, you may reduce from any problem that has been shown, in class or in CLRS, to be NP-complete.

6.046J / 18.410J Design and Analysis of Algorithms
Spring 2015