Practice Homework # 2

1. (Problem 7.6, Dasgupta et al.) Give an example of a linear program in two variables whose feasible region is infinite, but such that there is an optimum solution of bounded cost.

2. (Problem 7.7, Dasgupta et al.) Find necessary and sufficient conditions on the reals $p$ and $q$ under which the following linear program (a) is infeasible, (b) is unbounded, and (c) has a unique optimal solution.

   \[
   \begin{align*}
   \text{maximize} & \quad x + y \\
   \text{subject to} & \quad px + qy \leq 1 \\
   & \quad x, y \geq 0
   \end{align*}
   \]

3. (Problem 7.8, Dasgupta et al.) You are given the following points in the plane: (1,3), (2,5), (3,7), (5,11), (7,14), (8,15), (10,19). You want to find a line $ax + by = c$ that approximately passes through these points. (No line is a perfect fit).

   Write a linear program (you do not need to solve it) to find the line that minimizes the maximum absolute error, namely

   \[
   \max_{1 \leq i \leq 7} |ax_i + by_i - c|.
   \]

4. (Problem 7.13, Dasgupta et al.) In this simple two-player game, two players $R$ and $C$ each simultaneously choose an outcome, heads or tails. If both outcomes are equal, $C$ gives a dollar to $R$; if the outcomes are different, $R$ gives a dollar to $C$.

   (a) Represent the payoffs by a $2 \times 2$ matrix.

   (b) What is the value of this game, and what are the optimal strategies for the two players?

5. (Adapted from Kleinberg and Tardos, Problem 5, Chapter 6.) Apparently some languages (including Chinese and Japanese) are written without spaces between the words. Consequently software that works with text written in these languages must address the word segmentation problem—inferring likely boundaries between consecutive words in the text. If English were written without space, the analogous problem would consist of taking a string like “meetateight” and deciding that the best segmentation is “meet at eight” and not “me et at eight” or “meet a teight”, etc.

   How could this process be automated? One approach is to find a segmentation that maximizes the cumulative “quality” of its individual constituent words. Specifically, suppose that you have a black box that, given a string of letters $x = x_1 x_2 \ldots x_k$, returns a number $\text{quality}(x)$. The quality of real English words will be positive, while that of non-words will be negative.

   Given a long string of letters $y = y_1 y_2 \ldots y_n$, a segmentation of $y$ is a partition of its letters into contiguous blocks of letters; each block corresponds to a word. The total quality of a segmentation is the sum of the qualities of its blocks.

   Give an efficient algorithm that takes a string $y$ and computes a segmentation of maximum total quality.