CS515: Algorithms and Data Structures, Fall 2015

Homework 4*

Due: Mon, Nov/30/15

Homework Policy:

1. Students should work on homework assignments in groups of preferably three people. Each group submits to TEACH one set of typeset solutions, and hands in a printed hard copy in class. The hard copy will be graded.
2. You are allowed to discuss the homework with other groups, however, you must mention their names in your submission. Also, you must cite any other source that you use.
3. The goal of the homework assignments is for you to learn solving algorithmic problems. So, I recommend spending sufficient time thinking about problems individually before discussing them with your friends.
4. *I don't know policy:* you may write "I don't know" *and nothing else* to answer a question and receive 25 percent of the total points for that problem whereas a completely wrong answer will receive zero.
5. Algorithms should be explained in plain english. Of course, you can use pseudocodes if it helps your explanation, but the grader will not try to understand a complicated pseudocode.
6. More items might be added to this list. ☺

Problem 1. Suppose we are given an array $A[1 \cdots m][1 \cdots n]$ of non-negative real numbers. We want to round A to an integer matrix, by replacing each entry x in A with either $\lfloor x \rfloor$ or $\lceil x \rceil$, without changing the sum of entries in any row or column of A. For example:

$$\begin{bmatrix} 1.2 & 3.4 & 2.4 \\ 3.9 & 4.0 & 2.1 \\ 7.9 & 1.6 & 0.5 \end{bmatrix} \longmapsto \begin{bmatrix} 1 & 4 & 2 \\ 4 & 4 & 2 \\ 8 & 1 & 1 \end{bmatrix}$$

Describe and analyze an efficient algorithm that either rounds A in this fashion, or reports correctly that no such rounding is possible.

Problem 2.

(a) Give a linear-programming formulation of the maximum-cardinality bipartite matching problem. The input is a bipartite graph $G = (U \cup V, E)$, where $E \subseteq U \times V$; the output is the largest matching in G. Your linear program should have one variable for each edge.

(b) Now dualize the linear program from part (a).

*Problems are from Jeff Erickson's lecture notes. Looking into similar problems from his lecture notes on applications of maximum flow, linear programming and linear programming algorithms is recommended.

Problem 3. Given points $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ in the plane, the linear regression problem asks for real numbers a and b such that the line $y = ax + b$ fits the points as closely as possible, according to some criterion. The most common fit criterion is minimizing the ℓ_2 error, defined as follows:

$$\varepsilon_2 = \sum_{i=1}^{n} (y_i - (ax_i + b))^2.$$

But there are several other fit criteria, some of which can be optimized via linear programming.

1. The ℓ_1 error (or total absolute deviation) of the line $y = ax + b$ is defined as follows:

$$\varepsilon_1 = \sum_{i=1}^{n} |y_i - (ax_i + b)|.$$

   Describe a linear program whose solution (a, b) describes the line with minimum ℓ_1 error.

2. The ℓ_∞ error (or maximum absolute deviation) of the line $y = ax + b$ is defined as follows:

$$\varepsilon_\infty = \max_{i=1}^{n} |y_i - (ax_i + b)|.$$

   Describe a linear program whose solution (a, b) describes the line with minimum ℓ_∞ error.

Problem 4. Describe and analyze an algorithm that solves the following problem in $O(n)$ time: Given n red points and n blue points in the plane, either find a line that separates every red point from every blue point, or prove that no such line exists.