Analysis of Algorithms	**Take home midterm**
CS 6820 Fall 2014	**Due 5pm, Friday, October 24, 2014**

This problem set has 5 problems with parts of varying difficulty. I have assigned points to each part, with a maximum possible total of 100. A full solution for each problem includes proving that your answer is correct. If you cannot solve a problem, but can do some parts, or have partial results, write down how far you got, and why are you stuck.

For the midterm each student needs to work alone, and submit the solutions separately. The exam is open books, open notes, and you may use any fact we proved in class without proving the proof or reference, and may read the relevant chapters of the Kleinberg-Tardos or Kozen books, provided you state them clearly. However, you may **not use other published papers, or the Web to find your answer**.

Solutions can be submitted on CMS in pdf format (only). Please type your solution or write extremely neatly to make it easy to read. If your solution is complex, say more than about half a page, please include a 3-line summary to help us understand the argument.

Please direct all questions to Matvey and me. You can also ask clarifying questions using Piazza, where we will post all answers. Matvey and I will post special office hour schedule for the week.

(1) *(20 points)* Consider the coverage problem from problem set 2. Recall that the problem was defined by a bipartite graph where the two sides Y and X of the graph are interpreted as customers and stations respectively. We say that a subset of nodes in $A \subset Y$ can be served if and only if there is a matching M so that M matches all the nodes in A to nodes in X. We can then define the set of all matchable subsets of Y as $\mathcal{I} = \{A \subset Y : A$ can be served $\}$.

(a) *(10 points)* Show that the set system $\mathcal{I}$ defined above is a matroid.

(b) *(10 points)* As in the problem on the homework, assume that customers have signed up for different service levels of increasing cost, so that each customer $y \in Y$ pays $c_y \geq 0$ units of money, if served. Consider the following greedy algorithm for accepting customers for service: start with an empty matching M with no edges. Now consider customers in decreasing order of their service level (so customers with higher c_y values come first). When considering a customer y, accept y if there is an edge (x, y) that can be added to the matching M (without deleting or changing any previously added edges). Add y to the set A, and add $e = (x, y)$ to M. Does the resulting matching and accepted set of customers A maximize the total price, $\sum_{y \in A} c_y$, paid by them? Prove that it does, or provide a counterexample.

(2) *(15 points)* In a lot of numerical computations, we can ask about the "stability" or "robustness" of the answer. This kind of question can be asked for combinatorial problems as well; here's one way of phrasing the question for the Minimum Spanning Tree Problem.

Suppose you are given a graph $G = (V, E)$, with a cost c_e on each edge e. We view the costs as quantities that have been measured experimentally, subject to possible errors in measurement. Thus, the minimum spanning tree one computes for G may not in fact be the "real" minimum spanning tree.

Given error parameters $\varepsilon > 0$ and a specific edge $e' = (u, v)$, you would like to be able to make a claim of the following form:

($*$) Even if the cost of *each* edge were to be changed by at most ε (either increased or decreased), the edge e' would still not belong to any minimum spanning tree of G.

Such a property provides a type of guarantee that e' is not likely to belong to the minimum spanning tree, even assuming significant measurement error.

Give a polynomial-time algorithm that takes G, e', and ε, anddecides whether or not the property ($*$) holds for e'.

(3) *(15 points)* You are advising a growing startup that just outgrew their space in the building they have been for the last year. They are renting some additional space in a building a couple blocks away. The following principles have been agreed on for the move:

1. Headquarters needs to remain in the old building.
2. The company is arranged in groups, and no group should get split between two buildings.
3. There is one group who was most involved in negotiating rental of the new space, lets call them group A, they will move to the new rental.

Beyond these principles the move should be arranged to minimize total cost. While moving just group A is enough to have the rest fit in the old space, this may not be the optimal arrangement. Costs arise in a number of ways. If two groups collaborate a lot, there is a cost associated with braking them up across buildings. For each pair of groups (B, C), we have an estimate c_{BC} for the cost incurred if B and C are not in the same building. Further, space in the new building needs to be rented, and depending on the size of each group C, there is a cost r_C for renting space for group C in the new building if they move. Give a polynomial-time algorithm to find the minimum cost way to split groups across the two buildings.

(4) *(20 points)*

(a) *(10 points)* Consider the graph in the figure below. Give an optimal primal and dual solution for the maximum fractional matching problem on this graph (maximizing $\sum_e x_e$ for the constraints in part (b) without the c values in the objective function).

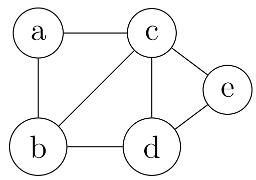

Figure 1: Graph for Problem 1.

(b) *(10 points)* Consider the following fractional maximum value matching, defined by the linear program below for a graph $G = (V, E)$ with $n = |V|$ vertices and $m = |E|$ edges, and the objective of maximizing the total weight of the fractional matching

$$x_e \geq 0 \text{ for all } e \in E$$
$$\sum_{e \text{ adjacent to } v} x_e \leq 1 \text{ for all } v \in V$$

$$\max \sum_e c_e x_e$$

What is the dual linear program to this linear program?

(5) *(30 points)* Statistical publications face the following rounding problem. When publishing statistics with really big numbers, it is better to round the numbers, as the resulting data table is more readable, and probably also more meaningful. On the other hand, rounding errors may accumulate, and result in very inaccurate data, which is not really OK. As an example consider the following table of data

County	*A*	*B*	*C*	*Total*
men ages 21-	11,618	7,611	6,712	25,941
women ages 21-	11,822	8,022	7,985	27,829
children ages 0-21	5,125	4,781	2,341	12,247
total	28,565	20,414	17,038	66,017

They would like to publish the data rounded to thousands. However, they face the following problem. Consider the first row, individually rounding each entry would result in the numbers 12K, 8K, and 7K, with a total of 27K, which really cannot be viewed as a rounded version of 25,941, being more than a thousand up from the the original value 25,941. We will say that a *proper* rounding of the table to multiples of M (e.g., $M = 1,000$) is one where each entry is an integer multiple of M, and the bottom row and the right column is the sum of the entries in the corresponding row and column respectively, and all entries (including all sums) are within less than M of the number original in the table (so as a consequence, numbers that happen to be integer multiples of M are not changed).

(a) *(5 points)* Consider the special case when $M = 1$ and all entries of the main part of the table are real numbers in the range $[0, 1)$ (with 0 possible, but 1 is not), and the last column and last row being the sum of the entries in the column or row respectively. Give a polynomial-time algorithm that finds a proper rounding if one exists.

(b) *(5 points)* Give a polynomial-time algorithm to find a proper rounding of any table with any value M, if one exist.

(c) *(10 points)* Show that the proper rounding required in (a) and (b) always exists.

(d) *(10 points)* The above definition of rounding is viewing a rounding proper with very high rounding error. For example, when $M = 1,000$, the rounding error is allowed to be as high as 999, almost M. How would your answer to (a–c) change if we modified the definition of a proper rounding of the table to require that the entries are within $\frac{2}{3}M$ of the original value?