Due on April 12th, 2016
105 points total

General Directions: If you are asked to provide an algorithm, you should clearly define each step of the procedure, establish its correctness, and then analyze its overall running time. There is no need to write pseudo-code; an unambiguous description of your algorithm in plain text will suffice.

All the answers must be typed, preferably using LaTeX. If you are unfamiliar with LaTeX, you are strongly encouraged to learn it. However, answers typed in other text processing software and properly converted to a pdf file will also be accepted. Before submitting the pdf file, please make sure that it can be opened using any standard pdf reader (such as Acrobat Reader) and your entire answer is readable. **Handwritten answers or pdf files that cannot be opened will not be graded and will not receive any credit.**

Finally, please read the detailed collaboration policy given on the course website. You are **not** allowed to discuss homework problems in groups of more than 3 students. **Failure to adhere to these guidelines will be promptly reported to the relevant authority without exception.**

Problem 1 (30 points)
Let $G = (V, E)$ be a unit-capacity graph with n vertices and m edges. In this question, we will revisit the randomized edge contraction algorithm for the global min-cut problem we saw in class. In this setting, the contraction algorithm will do one additional contraction so that the final graph is just a single vertex (instead of being left with two vertices which define a cut).

(a) (5 points) Show that in any run of the edge contraction algorithm, the edges contracted form a spanning tree of G.

(b) (15 points) Let $\mathcal{T}$ denote all the spanning trees in G. If we run the contraction algorithm, we will get a random spanning tree in $\mathcal{T}$ formed by the contracted edges, and we denote this distribution of spanning trees by $\mathcal{D}_1$. On the other hand, if we assign a random weight in $(0, 1)$ to each edge and compute a minimum spanning tree using Kruskal's algorithm, then we obtain another distribution $\mathcal{D}_2$ over $\mathcal{T}$. Show that these two distributions are identical.

(c) (5 points) Using the analysis we saw in class for the contraction algorithm, argue that there are $O(n^2)$ *global min-cuts* in any graph.

(d) (5 points) Observe that part (c) implies nothing about the number of s-t *min-cuts* there can be for two vertices $s, t \in V$. Give an example of a graph where there are $\omega(n^2)$ s-t min-cuts for a particular pair of vertices s and t. Recall that a function $g(n) = \omega(n^2)$ if $g(n)$ is *strictly* greater than n^2 asymptotically , i.e., $g(n) = \Omega(n^2)$ but $g(n) \neq \Theta(n^2)$ (so functions like n^3, n^{10}, 2^n, etc.).

Note that this counterexample needs to be constructed based on a general parameter n since we are attempting to show an asymptotic lower bound. This means you should describe a graph with n vertices, define the edge set, and then argue why there are $\omega(n^2)$ min-cuts.

Problem 2 (25 points)
Our friends Alice Algorithmix and Bob Bitfiddler are trying to learn randomized algorithms by playing a game. They start with n cards, all of which are initially placed in a single deck. The game proceeds in rounds. In each round, Alice chooses a deck containing at least 2 cards, secretly marks 1 card in the deck, and asks Bob to partition these cards into two decks of sizes chosen by her. For instance, if there are 10 cards in the deck chosen by her, she can ask Bob to partition the cards into two decks containing 8 and 2 cards respectively. (Bob does not know the marked card when partitioning the cards.) Once Bob has done the partitioning, his task is to record all the cards that were in the deck *not* containing the marked card. The game ends when every deck has exactly 1 card. The time that Bob takes to record the cards in a round is $\Theta(t)$, where t is the number of cards being recorded.

(a) (5 points) Show that if Bob uses a deterministic strategy for partitioning the cards, Alice can ensure that he will take a total time of $\Theta(n^2)$ over all rounds.

(b) (20 points) Instead, suppose Bob partitions the cards randomly. In other words, if he has to partition a deck of t cards into two decks of k and $t-k$ cards, then he shuffles the cards into a uniformly random order and chooses the first k cards in the first deck, and the last $t-k$ cards in the second deck. The random partitioning is done by Bob after Alice has marked a card from the overall deck. What is the expected time that Bob now takes over all rounds? (Give your answer using the $\Theta(.)$ notation.)

Problem 3 (25 points)
Your old friend Bob Bitfiddler is looking to open two new convenience stores in the small town of Linesville, NC. Linesville is composed of a single east-west street called Straight Street. There are n homes in total along Straight Street, and Bob has surveyed the residents of each of them in order to optimize the locations of his stores. For each house i located $\ell_i \in [0, 10]$ miles from the west gate of the town, his survey indicates there are r_i residents in home i, all of whom are willing to walk at most $d_i \in [0, 10]$ miles to shop at one of Bob's two new stores. Bob's goal is to pick the locations of his two stores $s_1, s_2 \in [0, 10]$ along Straight Street that maximize the total number of residents that are willing to visit at least one of the stores.

Formulate the above problem as an integer programming problem (i.e., the optimal solution to your IP should solve for the two locations that maximize the total number of shoppers). Also provide a brief justification for why your formulation is correct.

Some notes:

- Although s_1 and s_2 could be fractional locations (i.e, they need not be placed at mile $1, 2, \ldots$), you can still restrict other variables to be integers since we are asking for an integer programming formulation.
- As an intermediate step, use absolute values in your formulation. Then, try to rewrite the IP without absolute values.

Problem 4 (25 points)
This problem is about the minimum spanning tree (MST) problem in an undirected graph, where every edge has a non-negative cost.

(a) (15 points) Write an LP for the MST problem, which enforces the constraint that at least one edge is selected from every cut. Show that you can solve this LP in polynomial time by using an efficient separation oracle. (It is sufficient to identify the problem you need to solve in the separation oracle; you do not need to give an explicit algorithm.) Give an example to show that this LP is not integral.

(b) (10 points) Write a different LP for the MST problem where you enforce an upper bound on the number of edges selected from any induced subgraph, and a lower bound on the total number of edges. Show that any integral feasible solution to this LP is a spanning tree of the graph. (Recall that an LP is said to be integral if there is at least one optimal solution where all the variables take integer values.)