Due on April 26th, 2016
110 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 (20 points)
Let X be a set of n points in the plane. A point p in X is *Pareto-optimal* if no other point in X is both above and to the right of p. The Pareto-optimal points can be connected by horizontal and vertical lines into the *staircase* of X, with a Pareto-optimal point at the top right corner of each step.

(i) Describe an algorithm to compute the staircase of a given set of n points in the plane in $O(nh)$ time, where h is the number of Pareto-optimal points.

(ii) Describe an algorithm to compute the staircase of a given set of n points in the plane in $O(n \log n)$ time.

In both parts, you may assume that no two points have the same x- or y-coordinates. The output should be a linked list of Pareto-optimal points in left to right order.

Problem 2 (20 points)
Consider the following decision problems, SET COVER and VERTEX STEINER TREE:

SET COVER: You are given a set of n elements $U = \{x_1, \ldots, x_n\}$ and a collection of m sets $S = \{s_1, s_2, , \ldots, s_m\}$ such that each $s_i \subseteq U$ (i.e., each set in S is some subset of U). Given a fixed parameter K, determine whether it is possible to collectively include or *cover* all the elements in U with at most K sets from S. More formally, determine whether there exists a subset S' of S such that $\bigcup_{s \in S'} s = U$ and $|S'| \leq k$.

VERTEX STEINER TREE: You are given an unweighted, undirected graph $G = (V, E)$ and a set of terminal vertices $T \subseteq V$. Given a fixed parameter K, determine whether there exists a subgraph G' of G such G' connects all vertices in T (i.e., there exists a path between any $t_1, t_2 \in T$ in G') and includes at most K non-terminal vertices.

Given that SET COVER is NP-hard, prove that VERTEX STEINER TREE is NP-complete.

Problem 3 (10 + 15 points)
In the optimization version of the above problem, the objective is to find such a subgraph G' with minimum total node weight. First show that you can assume, without loss of generality, that all terminals are leaves of weight 0 in G.

We now describe a greedy algorithm for this problem. Define a spider to be a tree such that at most one vertex has degree greater than 2.

(i) Initially, let G_0 be the set terminals T with no edge.

(ii) At the i-th step, choose a subgraph S of G such that

- S is a spider
- all leaves of S are terminals

- S has the minimum cost-benefit ratio $\frac{C(S)}{B(S)}$. Here the cost $C(S)$ is the total node weight of S and the benefit $B(S) = \alpha(G_i) - \alpha(G_i \cup S)$, where $\alpha(G)$ denotes the number of connected components in graph G.

Let $G_{i+1} = G_i \cup S$. If G_{i+1} connects all terminals, stop the algorithm and output G_{i+1}; else go to ii.

Answer the following questions about this greedy algorithm.

(a) Give a polynomial time implementation of this greedy algorithm.

(b) Show that this algorithm returns an $O(\log \kappa)$-approximate solution, where $\kappa = |T|$.

(**Hint:** *Recall the analysis of the greedy algorithm for the set cover problem.*)

Problem 4 (5 + 15 points)
Consider the following problem. Given a set F of facilities and a set C of clients, we want to assign every client to one of the facilities. Let a_{ij} be the *connection cost* of assigning client c_i to facility f_j. Let b_j be the *opening cost* of facility f_j, that is, if f_j is assigned at least one client, this cost will be incurred (only once). The objective is to find an assignment of the clients to the facilities that minimizes the total cost.

(a) Show that the set cover problem is a special case of this problem.

(b) Show that the set cover problem is equivalent to this problem.

Problem 5 (15 + 10 points)
A maximal matching in an undirected graph is any matching (i.e., a subset of edges with disjoint end vertices) such that every edge in the graph is incident on at least one end vertex of an edge in the matching.

(a) Show that any maximal matching is a 2-approximation for the maximum matching of a graph.

(b) Now, give an algorithm to find the maximal matching of a graph in $O(m)$ time.