CS 577: Introduction to Algorithms	Homework 6
Out: 03/03/17	**Due: 03/10/17**

Ground Rules

- Review problems will be discussed at the following week's review.
- Self-graded problems will be discussed at the following week's review, and are to be self-graded by you during the review. Only students present at the review session and turning in their graded solutions in person will earn points.
- Graded problems are to be turned in on the due date at the beginning of the lecture.
- Self-graded problems should be done individually.
- Graded problems should be done in pairs.
- Please follow page limits for all self-graded and graded problems.
- Write your name(s) clearly on your submissions, and turn in each problem on **separate sheets of paper**.

Review Problems

1. Given a connected graph $G = (V, E)$ with weights w_e on edges and a spanning tree T, define the width of T to be the weight of the maximum weight edge in T.

   (a) Given a target width W, develop a linear time algorithm to determine whether G contains a spanning tree of width at most W.

   (b) Develop a linear time algorithm to find the smallest W such that there exists a spanning tree T with width W in G. You may assume that all edge weights are distinct.

   (Hint: Use your algorithm from part (a) as a subroutine, and do a sort of "binary search" over W.)

2. (Taken from "Algorithm Design") Let's consider a long, quiet country road with houses scattered very sparsely along it. (We can picture the road as a long line segment, with an eastern endpoint and a western endpoint.) Further, let's suppose that despite the bucolic setting, the residents of all these houses are avid cell phone users. You want to place cell phone base stations at certain points along the road, so that every house is within four miles of one of the base stations. Give an efficient algorithm that achieves this goal, using as few base stations as possible.

Self-graded Problems

3. **(Page limit: 1 sheet; 2 sides)** (Taken from "Algorithm Design") Suppose we have a connected graph $G = (V, E)$. Each edge e has a time-varying edge cost given by a function $f_e(t) = a_e t^2 + b_e t + c_e$ where $a_e > 0$, and $f_e(t)$ gives the cost of the edge at time t. Assume that these functions are positive over their entire range. Observe that the set of edges constituting the minimum spanning tree of G may change over time, and that the cost of the minimum spanning tree of G can be expressed as a function of the time t. Denote this function $c_G(t)$.

   Give an algorithm that takes the graph G and values $\{(a_e, b_e, c_e) : e \in E\}$ and returns a value of the time t at which the minimum spanning tree has minimum cost, $\min_t c_G(t)$. Your algorithm should run in time polynomial in the number of nodes and edges of the graph G, where you may assume that arithmetic operations on numbers can be done in constant time.

   (Hint: The set of edges in the MST depends on the ordering of edges by weight. At what times t does this ordering change?)

Graded Problems

4. **(Page limit: 1 sheet; 2 sides)** Solve Problem 4, **Bottleneck distance**, on Page 9 of Chapter 20 in the book "Algorithms, Etc." Your algorithm should run in polynomial time and should return a two-dimensional array with the bottleneck distances.

   Link: `http://jeffe.cs.illinois.edu/teaching/algorithms/notes/20-mst.pdf`.

5. **(Page limit: 2 sheets; 3 sides)** Suppose that on a busy day of the semester you get homework for each of the n courses that you are taking. Homework for the i^{th} course is due within t_i days, and you earn p_i points for submitting the homework on time. Unfortunately it takes you one full day to do each homework. So, for example, if you have 4 homeworks due within 2, 3, 1, and 2 days, respectively, you cannot submit all four on time, but you can submit the first three on time by doing the third one on day 1, the first one on day 2, and the second one on day 3. Your objective is to maximize the total number of points you can earn by submitting a subset of the homeworks on time.

   (a) Let S be a subset of all homeworks. Call S "reasonable" if there is an order in which you can do the homeworks in S so that each one can be submitted on time. Prove that S is reasonable if and only if for all integers $t \leq n$, the number of homeworks in S due within t days or less is no more than t.

   (Hint: Consider doing the homework in increasing order of due date.)

   (b) Design a polynomial time greedy algorithm to determine which homework to do on which day so as to maximize the number of points you can earn by submitting a subset of the homework on time. Prove the correctness of your algorithm and analyze its running time.

   (Hint: Build up a reasonable set of homeworks greedily by considering homeworks in a particular order and adding each homework to your schedule if the set of selected homeworks continues to remain reasonable. Use part (a) to check for feasibility. What order should you consider homeworks in to maximize the number of points earned? Use an exchange argument to prove the correctness of your algorithm.)