Due: Wednesday, March 2, beginning of lecture

NOTE: Each problem set only counts 5% of your mark, but it is important to do your own work (but see below). These assignments will be followed by term tests, each worth 15% of your final grade. You may consult with others concerning the general approach for solving problems on assignments, but you must write up all solutions entirely on your own. You may chose to work in pairs and then submit one assignment but both partners should work on all questions. Anything else is *plagiarism*, and is subject to the University's Code of Behavior. You will receive 1/5 points for any question/subquestion for which you say "I do not know how to answer this question". You will receive .5/5 points if you just leave the question blank.

1. Let $G = (V, E)$ be a directed graph with edge costs $c : E \rightarrow \Re$. Assume that all directed cycles in G have positive cost.

   (a) Describe a dynamic programming algorithm to compute the number of different simple min cost $s - t$ paths in G. Here different just means that the paths are not identical. As in the previous DP algorithm question, define an appropriate semantic array A , and a recurrence (including the base cases) for computing elements of A. Briefly justify why the recurrence is correct.

   (b) Could your algorithm be modified to count the number of different simple (not necessarily min cost) $s - t$ paths in G?

2. Consider the situation where we are given an array L of n elements which cannot be sorted (e.g. the elements are complex numbers) but there is a test for equality of any two elements.

   Describe an $O(n \log n)$ divide and conquer algorithm to find the set of *all* elements (if any) that occur at least $\lceil n/3 \rceil$ times in L.

   Give a recurrence that describes the number of pairwise comparisons made by your algorithm.

3. We will discuss in class the problem of multiplying two degree n polynomials using $O(n^{log_2 3})$ arithmetic operations which is approximately, $O(n^{1.59})$. Using this result (as a subprogram), describe a divide and conquer algorithm using $O(n^{log_2 3})$ arithmetic operations which given $\{a_1, \ldots, a_n\}$ computes the (coefficients of the) polynomial $p(x) = \Pi_{1 \leq i \leq n}(x - a_i) = (x - a_1)(x - a_2) \cdots (x - a_n)$. Give a recurrence that describes the number of arithmetic operations made by your algorithm and briefly show why this recurrence yields the desired bound. You may assume $n = 2^k$ for some k.

4. Suppose we are given a flow network $\mathcal{F}$ with edge capacities $\{c_e | e \in E\}$ and we are also given a max flow f in $\mathcal{F}$. Let e_1 and e_2 be two different edges in E and form a new flow network $\mathcal{F}'$ by setting $c'_{e_1} = c_{e_1} - 1$ and $c'_{e_2} = c_{e_2} + 1$ and $c'_e = c_e$ for all other $e \in E$.

   Show how to compute a max flow f' in $\mathcal{F}'$ in time $O(|E|)$.

5. Possibly more to follow