CS 38 Introduction to Algorithms	**Spring 2014**

Midterm

Out: April 29 *Due:* **May 6**

This is a midterm. You may consult only the course notes and the optional text (CLRS). *You may not collaborate.* The full honor code guidelines can be found in the course syllabus.

There are 5 problems on 2 pages. Please attempt all problems. **To facilitate grading, please turn in each problem on a separate sheet of paper and put your name on each sheet. Do not staple the separate sheets.** Good luck!

1. Given an undirected graph $G = (V, E)$ in adjacency list format, we are interested in determining whether or not it contains a triangle; i.e., three vertices a, b, c for which $(a, b), (b, c)$, and (a, c) are all edges.

   (a) One idea is to use a BFS traversal, and when we are currently at vertex u, if we see an edge (u, v) to an already-explored vertex v, we say there is a triangle if the predecessor of u in the BSF tree and the predecessor of v in the BFS tree are the same. Give a small counterexample showing that this idea fails.

   (b) Give an algorithm that runs in time $O(n^\alpha)$ for some constant $\alpha < 3$. Hint: the intended solution requires you to work with the adjacency *matrix* A of the graph. What does entry (i, j) of A^2 tell you?

2. Given a weighted, connected, undirected graph $G = (V, E)$ and a subset $F \subseteq E$ of "mandatory edges", design an algorithm to find the spanning tree of G that contains F, of minimum cost among all such trees. Do this via the following two steps.

   (a) Show that for a matroid $\mathcal{I}$ on universe E, and an element $e \in E$, the set system on $E' = E \setminus \{e\}$ defined by

   $$\mathcal{I}' = \{A : (A \cup \{e\}) \in \mathcal{I}\}$$

   is a matroid.

   (b) Argue that the subsets of G's edges that, together with F, form spanning trees, constitute the bases of a matroid. Show that you can find the minimum cost such tree in time $O(m \log m)$ by assigning weights and implementing the greedy algorithm to find a maximum weight independent set in this matroid.

3. Given three lists of integers $a_1, a_2, \ldots, a_n$, $b_1, b_2, \ldots, b_n$, and $c_1, c_2, \ldots c_n$, we are interested in determining whether there exist i, j, k such that $a_i + b_j + c_k = 0$.

   (a) Show how to solve this problem using $O(n^2 \log n)$ operations (you may count a comparison or addition operation between two integers from the lists and copying/moving an integer from the lists to be single operations, even though the integers may be large).

(b) Suppose we know that all of the integers lie in the range $[-M, M]$. Show how to solve this problem using $O(M \log M)$ operations.

Hint: for both parts, use binary search; for the second part, use, in addition, fast polynomial multiplication. Given a set S, your polynomials will have the form $f(X) = \sum_{i \in S} X^i$.

4. Given n points in the plane, we want to find the lightest triangle, i.e., the triple of points a, b, c such that $d(a, b) + d(b, c) + d(c, a)$ is minimum, where $d(\cdot, \cdot)$ is the Euclidian distance function. Give a divide-and-conquer algorithm for this problem that runs in time $O(n \log^2 n)$.

5. Given a directed graph, a Hamilton cycle in G is a cycle that visits every vertex exactly once. You may recall that the problem of finding a Hamilton cycle is NP-complete, so we do not expect a polynomial time solution. The brute-force algorithm which tries all orderings of the n vertices takes $n!$ time.

   Give a dynamic programming algorithm that decides if G contains a Hamilton cycle, and that runs in time $O(n^2 2^n)$ (this time bound is much better than $O(n!)$). Hint: your dynamic programming table will be indexed by pairs (S, v) where S is a subset of vertices, and v is a vertex. Subproblem (S, v) is to determine if there is a path starting at vertex x (a fixed vertex) and passing through all of S, ending at v.