Homework assignment 3, due Tues Oct. 30 before class (for FEEDS students: due THURS NOV 1)

–Look up course webpage under "Homeworks," specifically for
late submission policy which you must follow strictly – otherwise you could lose all points;
–Cheating policy and bonus policy under "about homeworks."
–Please make sure to include your code when you submit your homework, or risk losing points.
Most answers should contain at least the following 4 parts. (Certain problems may require more, and a very few, less).
- The idea of the algorithm – you could choose to illustrate this with an example. Mention familiar design techniques used, if any.
- A pseudocode of the algorithm. Any segment of code that takes constant time may be abbreviated by one instruction. Any algorithm done in class or in the assigned readings can be quoted. All else must be explained.
- A proof of correctness: argue why the desired output is produced when the algorithm halts, and that it does halt.
- An analysis of the worst case time complexity, in terms of the relevant input size parameters.

1. The adventurous bicyclist problem is the following. The input is a map with n mountainous villages; for each pair of villages, the maximum altitude that occurs in a trail between them; and the probability (a number between 0 and 1) that there is no landslide blocking the trail; and a designated pair of cities, s and t. The outputs required are: a route from city s to city t, which minimizes the maximum altitude occuring on the route; and a different route from s to t that minimizes the probability that there is a landslide blocking some trail on the route. Give an efficient algorithm to output these two routes.

   Hint: Both output routes require a simple adaptation of the shortest path algorithm, and both require a careful argument as to why the adaptation works. Here's all the probability knowledge you need: minimizing the probability that there is a landslide blocking some trail on a route q is the same as maximizing $good(q)$, where $good(q)$ is the probability that there is no landslide blocking any trail on the route q. Note that $good(q)$ is just the product - over the trails t on the route - of the (input) probability that t is not blocked by a landslide.

2. (20 points, compulsory) Professors in the CISE dept are required to join various committees, and naturally, there is the chairperson, whose job is to take the list of all the committees and find a representative for each committee, subject to the following constraints:
   Each committee has exactly one representative, and no one shall be a representative of more than one committee.

   The chairperson has appointed you to solve the problem efficiently: your algorithm should take as input the list of all committees and their members, and your algorithm should produce a legal list of representatives, or report that none exists.. Restate it as a network flow problem, and show that your restatement is is correct.

3. (20 points, compulsory) Design a fast algorithm, and prove correctness and time-complexity of the following problem specification.
   A *triangulation* is a set of points (given as (x, y) coordinates), and straight line segments between pairs of points, such that no two line segments intersect, and moreover, it is impossible to connect any more pairs of points without creating intersections. (Draw an example to figure out this definition–notice the triangles that appear).
   Give the fastest algorithm that you can think of, which, given a triangulation as input, together with a source point s, outputs the shortest paths from s to each of the other points using only edges in the triangulation. The length of the path is the sum of the (Euclidean) lengths of the edges on the path. Hint: First use the triangle inequality (the sum of the lengths of any two edges of a plane triangle is strictly greater than the length of the third edge) to design and prove correctness of the algorithm. Think in terms of other algorithms besides the usual shortest path algorithm. Then, in the analysis of efficiency, use the fact that no two roads cross to give a good bound on $|E|$ as a function of $|V|$, and give your complexity entirely in terms of $|V|$.

4. (20 points, compulsory) Design an algorithm for the following problem.
   Input: The x and y coordinates of two sets, S_1 and S_2, of points on the plane. Both sets have n points.
   Output: A 'yes' if there is a straight line seperating S_1 from S_2, and a 'no' otherwise.
   The more time efficient your algorithm, the more points you get.

5. (Bonus question)

(a) Let $G = (V, E)$ be a biconnected undirected graph (one needs to remove at least 2 vertices to disconnect the graph). Give an $O(|E|)$ algorithm to assign a direction to each edge in E so that the resulting directed graph is strongly connected (there is a path *from* every vertex *to* every other vertex).

(b) An undirected graph $G = (V, E)$ is *transitively orientable* if there is an orientation of its edges such that the resulting directed graph $D(G) := (V, A)$ is transitive, i.e $(u, v) \in A$ and $(v, w) \in A$ implies $(u, w) \in A$. Design an $O(|E|)$ algorithm for the following problem. Given a transitively orientable graph $G = (V, E)$ and a transitive orientation $D(G) = (V, A)$, color the vertices of G with the *minimum* number of colors such that no two adjacent vertices in G are assigned the same color.

Hint: Note that $D(G)$ is acyclic. Why? Define the height function h on V as follows: $h(v) = 0$ if v is a sink in $D(G)$; otherwise, $h(v) = 1 + max\{h(w) : (w, v) \in A\}$.