Algorithms
Computer Science 140 & Mathematics 168
Instructor: B. Thom
Fall 2004
Homework 9b
Due on Tuesday, 11/02/04 (beginning of class)

1. **[25 Points] Minimum Spanning Trees from the Shortest Paths Perspective!**

   Show that a minimum spanning tree (MST) of an undirected graph is equivalent to the *bottleneck shortest paths* tree (BSPT) for the graph. A BSPT is a tree that spans the graph that has the following property: For every pair of vertices u and v, it contains a path connecting them whose longest edge is as short as possible.

2. **[25 points] Negative-Cost-Finding With Floyd-Warshall!**

   In this problem, you will show how to modify Floyd-Warshall so that it can detect the presence of a negative-cost cycle (NCC).

   (a) Describe your modification. Rather than writing pseudocode, just describe in English what your cycle detection criterion is. This criterion should amount to a claim of some sort, e.g. some condition *foo* is **equivalent** to the graph having an NCC.

   (b) Demonstrate your algorithm's correctness by carefully proving this claim (as we did for Bellman-Ford in class, you must prove both directions).

3. **[25 points] An Algorithmic Get-Rich-Quick Scheme!**

   Your job at My-I'm-Soft was fun for awhile, but your reputation has gotten out and you've been hired back by the brokerage firm of Weil, Proffet, and Howe at a whopping salary. (We won't be specific about the actual amount, which is to say that your very happy that the company writes its payroll checks on a 64-bit computer!).

   Weil, Proffet, and Howe has just entered the **arbitrage** business. Arbitrage is a money-making scheme involving anomalies in international currency exchange rates. For example, imagine that 1 U.S. dollar buys 0.8 Zambian kwachas, 1 Zambian kwacha buys 10 Mongolian tughriks, and 1 Mongolian tughrik buys 0.15 U.S. dollars. Then, by converting currencies, a trader can start with 1 U.S. dollar and buy $0.8 \times 10 \times 0.15 = 1.2$ U.S. dollars. By capitalizing on such anomalies quickly (before they're detected and corrected by the markets), huge amounts of money can be made.

   We'll assume that we're given n currencies and the exchange rate between every pair of currencies. That is, we are given currencies $c_1, \ldots, c_n$ and an $n \times n$ table R such that one unit of currency c_i buys $R[i, j]$ units of currency c_j. (The values for all the diagonal elements are one; if one were to exchange a currency with itself one gains or looses nothing. Further, negative values for $R[i, j]$ never occur.)

   (a) Lets begin with the following problem. Given a list of currencies and exchange rates and a particular currency c_i—i.e. a single source—determine the maximum amount of each currency that you can obtain, beginning with 1 unit of currency c_i. For this part of the problem, assume that there exist no cycles that allow you to get arbitrarily rich via arbitrage.

i. Describe how to modify Bellman-Ford so that it can perform this task. Your algorithm can be described in English (pseudo-code is not necessary).

ii. Describe in a sentence or two why your algorithm is correct. A detailed proof (like one seen for Bellman-Ford in class) is not needed. Rather by identifying what invariant is maintained, you should be able to appeal to our argument in class.

iii. Analyze your modified algorithm's runtime (again, borrowing results from class is fine).

(b) Now assume that the exchange rates are such that it may be possible to get rich via arbitrage. That is, there may exist a cycle of currency exchanges that allows you to make more of your starting currency than you had initially.

i. Describe how you could modify Bellman-Ford to determine if it is possible to profit from arbitrage with the given set currencies and exchange rates (again, pseudo code is fine).

ii. Carefully prove that your modified algorithm is correct. Don't rely on results from class here, rather reconstruct the proof shown in class so that it applies to this modified setting.

(c) Hopefully this problem has convinced you that being able to print out some NCC in a graph when on exists is a pretty powerful thing to be able to do! However, we won't ask you to come up with such a procedure in this homework. You might wish to think about how one might do this, though. If there is enough interest, we can discuss ways to do this in class.