CSC364 Summer 2004 — Homework 3

The following questions are assigned each week. You need only hand in those questions marked with a star * as part of your assignment, and only those questions will be marked. Make sure you include a completed and signed cover page for each problem.

Week 9

*1. (hand in) Consider the All-Pairs-Shortest-Paths (APSP) problem as defined in the lecture notes:
Input: A directed graph G and edge cost function $c : E \to \mathbb{R}^{\geq 0} \cup \infty$.
Object: Find the minimum cost path between u and v for every pair of vertices u and v.

Now consider the related problem All-Pairs-Longest-Paths (APLP):
Input: A directed graph G and edge cost function $c : E \to \mathbb{R}^{\geq 0} \cup \infty$.
Object: Find the maximum cost simple (no cycles) path between u and v for every pair of vertices u and v.

Consider the Floyd-Warshall algorithm for solving APSP. Is there a *simple* modification of the Floyd-Warshall algorithm which will solve APLP? (A simple modification will change the mathematical operations but not the structure of the dynamic programming algorithm.) Either give the required modification or explain why it is not possible.

2. Returning to decision problems, define the LONGEST-PATH decision problem:
Input: A directed graph G, two vertices u, v and an integer k.
Question: Does there exist a simple (without loops) path from u to v in G with at least k edges?

Define the optimization problem LONGEST-PATH-LENGTH:
Input: A directed graph G and two vertices u, v.
Object: Return the maximum length of a simple path between u and v in G.

Show that the optimization problem LONGEST-PATH-LENGTH can be solved in a worst case time complexity which is a polynomial of its input size if and only if the decision problem LONGEST-PATH can be solved in a worst case time complexity which is a polynomial of its input size. (You may use question 1 of the previous homework in your justification.)

Week 10

3. In a graph G, a Hamiltonian path is a simple path that includes every vertex of G. A Hamiltonian cycle is a simple cycle that includes every vertex of G.

   Defive the decision problems as:
   HP: given a graph G, does G contain a Hamiltonian path?
   HC: given a graph G, does G contain a Hamiltonian cycle?

   Prove both HC $\leq_p$ HP and HP $\leq_p$ HC.
   Hint: It is not as easy as it looks!
   (We use $\leq_p$ as shorthand for $\leq_m^p$.)

4. Prove HC and HP are both in NP.

*5. (hand in) Consider a graph $G = (V, E)$. An *independent set* of G is a subset of the vertices $I \subseteq V$ such that for each pair of vertices in I there is no edge of G connecting that pair. A *clique* of G is a subset of the vertices $K \subseteq V$ such that for every pair of vertices in K there is an edge of G connecting that pair.

   Let INDEPENDENT SET be the decision problem:
   Instance: $\langle G, B \rangle$ where G is an undirected graph and B is an integer.
   Acceptance Condition: Accept if and only if G has an independent set of size at least B.

   Let CLIQUE OPT be the search problem:
   Instance: $\langle G \rangle$ where G is an undirected graph.
   Output: A maximum clique of G.

   Prove that CLIQUE OPT $\xrightarrow{p}$ INDEPENDENT SET. That is, given a polynomial time algorithm for the INDEPENDENT SET decision problem, we can find a maximum clique of G in polynomial time.

Week 11

Note that there are two hand in problems this week!

*6. (hand in) Define the SUBGRAPH-ISOMORPHISM decision problem as follows:
Given: graphs $G = (V_1, E_1)$ and $H = (V_2, E_2)$.
Decide: does G contain a subgraph isomorphic to H?
(i.e., are there subsets $V \subseteq V_1$ and $E \subseteq E_1$ such that $|V| = |V_2|$, $|E| = |E_2|$, and there exists a one-to-one mapping $f : V_2 \to V$ satisfying $\{u, v\} \in E_2$ iff $\{f(u), f(v)\} \in E$?)

Prove that SUBGRAPH-ISOMORPHISM is **NP**-complete.
Hint: Consider one of the problems CLIQUE, HP, or HC.

7. Complete the proof for Theorem 7 in the on-line notes (pages 11–12) that PARTITION is **NP**-complete.

*8. (hand in) Ontario has recently deregulated its energy market. This new "free" market works by matching producers of energy (power plants) with consumers of energy (a public utility, a factory). Each producer makes a *sell* order indicating the price for which the producer will sell energy (in cents per kilowatt) and the number of kilowatts the producer will sell. Each consumer makes a *buy* order indicating the price for which the consumer is willing to purchase energy and the amount the consumer wishes to purchase. If the price of a buy order is higher than or equal to the price on a sell order, a sale takes place and the consumer then sells x kilowatts to the consumer where x is the minimum of the amount the consumer will buy or the producer will sell.

The Ontario Energy Board approaches you with the following problem. Consumers and producers want the ability to create a "secret" order. A secret order indicates that the producer (or consumer) is willing to sell (buy) all of its energy at a lower (higher) price, but only if they are guaranteed to sell (buy) a minimum amount. For example, suppose Ontario Hydro currently has a sell order on the market stating that it is willing to sell up to 2,000,000 kilowatts at 4 cents per kilowatt. However, Ontario Hydro is also willing to sell at 3 cents a kilowatt if it can sell at least 1,750,000 kilowatts.

The problem Ontario Energy Board has is that it wants the program to automatically match secret orders. That is, if there are enough sellers of energy who are selling at or below the secret buy price of an order (including any secret sell prices) and the total amount is at least the minimum set by the secret buy, then we have a match. However, if the match includes some secret sell orders, then we may also need to include some other orders on the buy side so that the secret sell gets its minimum.

Consider the following example.

Seller	Price	Amount	Secret Price	Secret Min
Bill's Wind Farm	3 ¢	250,000	—	—
Ontario Hydro	4 ¢	2,000,000	3 ¢	1,750,000

Buyer	Price	Amount	Secret Price	Secret Min
Ford Motors	2.5 ¢	1,500,000	3 ¢	1,000,000
Toronto Hydro	2 ¢	1,500,000	3 ¢	1,200,000

If we did not have secret orders, no sales would occur in this example because the producers are asking for a higer price than the consumers want to spend. However, if we look at the secret price, then we have a sale.

Ontario Hydro will sell at 3 cents if it can sell at least 1,750,000 kW. This minimum can be met if we sell to both Ford Motors and Toronto Hydro. However, Ford Motors and Toronto Hydro will only buy at 3 cents if they can get, combined, 2,200,000. Ontario Hydro is only selling 2,000,000, but if we include Bill's Wind Farm, then we have a total amount of 2,250,000 kW which satisfies all the minimum quantities. Thus, Bill's Wind Farm and Ontario Hydro will sell a total of 2,250,000 kW at 3 cents per kW to Ford Motors and Toronto Hydro.

Here are the specifications of the algorithm that Ontario wants you to design. The input to the algorithm will be a list of buy and sell orders, each with at most one secret price. The output is a series of sales.

You are to either design an algorithm or prove the decision version of the problem is **NP**-complete. If an algorithm is possible, will it be feasible for the Ontario market? The market must be real-time so any algorithm which is worse than $O(n^2)$ will not be feasible.