Due Date: April 9, 2013

Problem 1: A subsequence is obtained by extracting a subset of elements from a sequence, but keeping them in the same order; the elements of the subsequence need not be contiguous in the original sequence.

Call a sequence \(X[1..k]\) of numbers oscillating if \(X[i] < X[i + 1]\) for all even \(i\), and \(X[i] > X[i + 1]\) for all odd \(i\). Describe an efficient algorithm to compute the length of the longest oscillating subsequence of an arbitrary array \(A\) of \(n\) integers.

Problem 2: Suppose we are given a sequence of undirected graphs \(G_0, G_1, G_2, \ldots, G_b\) with the same node set \(V\) and different edge sets \(E_0, E_1, E_2, \ldots, E_b\); that is, \(G_i = (V, E_i)\) for \(i = 0, 1, \ldots, b\). Assume every \(G_i\) is connected.

Now consider two particular nodes \(s, t \in V\). For an \(s\)-\(t\) path \(P\) in one of the graphs \(G_i\), we define the length of \(P\) to be the number of edges in \(P\), denoted as \(\ell(P)\). Our goal is to produce a sequence of paths \(P_0, P_1, \ldots, P_b\), where \(P_i\) is an \(s\)-\(t\) path in \(G_i\), such that each path is relatively short and there are not too many changes. Formally, we define \(\text{changes}(P_0, P_1, \ldots, P_b)\) to be the number of indices \(i (0 \leq i \leq b - 1)\) for which \(P_i \neq P_{i+1}\).

Fix a constant \(K > 0\). We define the cost of the sequence of paths \(P_0, P_1, \ldots, P_b\) to be

\[
\text{cost}(P_0, P_1, \ldots, P_b) = \sum_{i=0}^{b} \ell(P_i) + K \cdot \text{changes}(P_0, \ldots, P_b).
\]

(i) Suppose it is possible to choose a single path \(P\) that is an \(s\)-\(t\) path in each of the graphs \(G_0, \ldots, G_b\). Give a polynomial-time algorithm to find the shortest such path.

(ii) Give a polynomial-time algorithm to find a sequence of paths \(P_0, \ldots, P_b\) of minimum cost, where \(P_i\) is an \(s\)-\(t\) path in \(G_i\) for \(i = 0, \ldots, b\).

Problem 3: Integer linear programming (ILP) is the same as linear programming except that one considers only integer solutions, i.e.,

\[
\begin{align*}
\min & \quad c^T x \\
\text{s.t.} & \quad Ax \leq b \\
& \quad x \geq 0, x \in \mathbb{Z}
\end{align*}
\]

The vertex cover of a graph \(G = (V, E)\) is a subset \(C \subseteq V\) of vertices so that each edge in \(E\) is incident to at least one of the vertices in \(C\).

(i) Show that the problem of computing the minimum-size vertex cover can be formulated as an instance of ILP.
(ii) Suppose we relax the ILP in (i) to an LP by removing the integer constraint $x \in \mathbb{Z}$. Show that there is a graph for which the optimal LP solution is not integral (i.e., values of some of the variables are not integers).

**Problem 4:** Write the dual to the following linear program.

$$\begin{align*}
\max & \quad x_1 + x_2 \\
2x_1 + x_2 & \leq 3 \\
x_1 + 3x_2 & \leq 5 \\
x_1, x_2 & \geq 0
\end{align*}$$

Find the optimal solutions to both primal and dual LPs using the simplex method. For the primal, you can use $x_1 = 0$ and $x_2 = 0$ as the initial basic feasible solution (BFS), and for the dual, you can use $y_1 = 1$ and $y_2 = 0$ as the initial BFS, where $y_1$ and $y_2$ are the dual variables associated with the first and second constraints in the primal, respectively.

**Problem 5:** (Matching pennies) In this simple two-player game, the players (call them $R$ and $C$) each choose an outcome, heads or tails. If both outcomes are equal, $C$ gives a dollar to $R$; if the outcomes are different, $R$ gives a dollar to $C$.

(i) Represent the payoffs by a $2 \times 2$ matrix.

(ii) What is the value of this game? Compute the optimal strategies for the two players using the simplex algorithm.