CSC364 **Homework 3** **Winter 2002**

This problem set is worth 5% of your mark, and it is important to do your own work. You must write up all solutions entirely on your own. Anything else is *cheating* and is subject to prosecution under the University's Code of Behavior. This homework is due at the beginning of tutorial on Wednesday, April 10. Late assignments will receive a grade of 0.

1. Prove Lemma 1 of the lecture notes on Greedy Algorithms.

2. We are given m jobs to process. Job i requires t_i hours to process and has deadline d_i; that is, job i should be completed within d_i hours after time 0. A *schedule* is an ordering for the jobs. If the jobs are processed sequentially starting at time 0 in the order specified by a schedule S, then each job i will be completed at some time c_i hours after time 0. The penalty p_i for job i under schedule S is $p_i = 0$ if $c_i \leq d_i$, and $p_i = c_i - d_i$ if $c_i > d_i$. The total penalty is $P = p_1 + ... + p_m$.

   (a) Give a simple greedy algorithm (based on deadlines) for finding a schedule S such that if there exists any schedule with total penalty $P = 0$, then schedule S results in total penalty $P = 0$.

   (b) Prove that your greedy algorithm has the property specified above. Do this by following the standard scheme for proving correctness of greedy algorithms (see for example, the proof of correctness of Kruskal's algorithm).

3. (a) Suppose that the edge costs are all distinct in an input G to the MST problem. Prove that G has a unique minimum cost spanning tree. Do this by considering the *proof* that Kruskal's algorithm always finds a minimum cost spanning tree.

   (b) Now suppose that all edge costs are distinct, except $c_1 = c_2$, where $c_1 = c(e_1)$ and $c_2 = c(e_2)$ are the two minimum edge costs. Prove or give a counter-example: G has a unique minimum cost spanning tree.

   (c) Finally suppose that all edge costs are distinct, except $c_1 = c_2 = c_3$, where $c_1 = c(e_1), c_2 = c(e_2), c_3 = c(e_3)$ are the three minimum edge costs. Prove or give a counter-example: G has a unique minimum cost spanning tree.

4. We can use dynamic programming on a directed graph $G = (V, E)$ for speech recognition. Each edge $(u, v) \in E$ is labelled with a sound $\sigma(u, v)$ from a finite set Σ of sounds. The labelled graph is a formal model of a person speaking a restricted language. Each path in the graph starting from a distinguished vertex $v_0 \in V$ corresponds to a possible sequence of sounds produced by the model. The label of a directed path is the concatenation of the labels of the edges on that path.

   (a) Describe an efficient dynamic programming algorithm that, given an edge-labelled graph G with distinguished vertex v_0, and a sequence $s = < \sigma_1, \cdots, \sigma_k >$ of characters from Σ, returns a path in G that begins at v_0 and has s as its label, if any such path exists. Otherwise the algorithm should return NO-SUCH-PATH.

   (b) Prove that your algorithm is correct

   (c) Analyze the running time of your algorithm.