Algorithms, 2011 Fall, Homework 2
(Due: OCT 10)

September 27, 2011

Problem 1: (KT pp.189, problem 3) You are consulting for a trucking company. Each truck have a fixed limit $W$ on the maximum amount of weight it can carry. Boxes arrive one by one and each box $i$ has a weight $w_i$. The trucking station is very small and only one truck can be at the station at any time. Company policy requires that boxes are shipped in the order of they arrive. Prove the following greedy algorithm uses the minimum number of trucks: they pack boxes in the order they arrive, and whenever the next box does not fit, they send the truck on its way.

Problem 2: (KT pp.192, problem 8) Suppose you are given a connected graph $G$, with edge costs that are all distinct. Prove $G$ has a unique minimum spanning tree.

Problem 3: (KT pp.200, problem 22) If the edges cost in a graph are not all distinct, there might be many minimum spanning trees. Suppose you have a spanning tree $T$ such that each edge in $T$ belongs to some minimum spanning tree. Can we conclude $T$ itself must be a minimum-cost spanning tree? Give a proof or a counterexample.

Problem 4: (Minimum Bottleneck Spanning Tree) For a spanning tree $T$ of a graph $G$, we define the bottleneck edge of $T$ to be the most expensive edge in $T$. Design a polynomial algorithm to find a spanning tree of $G$ such that the bottleneck edge is as cheap as possible.

Problem 5: Design a polynomial time algorithm for the following problem. You have a directed graph with two special vertices $s$ and $t$. Each edge $e$ has a positive capacity $c(e)$ and a positive delay $d(e)$. The time to ship $\delta$ units of data from $s$ to $t$ on path $P$ is $\sum_{e \in P} d(e) + \frac{\delta}{C}$ where $C = \min_{e \in P} c(e)$. You task is to find a path such that the time to ship $\delta$ units of data from $s$ to $t$ is minimized.

Problem 6: (Periodic Scheduling) There are $n$ tasks. Each task takes one unit of time of perform and the machine can process at most 1 task in each time slot. The requirement is that task $i$ should be scheduled once in each time period $p_i$. For example, if task $i$ has period $p_i$, we must schedule it in each time-window of the form $[p_i j + 1, \ldots, p_i (j+1)]$. Let $L = LCM(p_1, p_2, \ldots, p_n)$ (LCM stands for least common multiple). Please answer the following questions.

1. Prove that if $\sum_i \frac{1}{p_i} > 1$, there is no periodic schedule (i.e., you can not find a schedule for the first $L$ time slots).
2. Suppose $\sum_{i} \frac{1}{p_i} \leq 1$. Prove the following greedy algorithm can find a feasible schedule for the first $L$ time slots: At each time slot, schedule the unfulfilled task with the nearest deadline (if there are several, do it in arbitrary order).

**Problem 7:** (KT pp.202, problem 26) Suppose each edge of the graph $G$ is associated with a cost function $f_e(t) = a_et^2 + b_et + c_e$ where $a_e > 0$ (we can think $t$ is the time parameter and $f_e(t)$ is the length of $e$ at time $t$). We are given the graph $G$ and the values $\{a_e, b_e, c_e\}$ as input. Give a algorithm that returns a value $t$ at which the minimum spanning has a minimum cost (over all $t$). (We may assume that arithmetic operation on $\{a_e, b_e, c_e\}$ can be done in constant time per operation.)

**Problem 8:** You have a graph $G$. Each edge $e$ has two cost $A_e$ and $B_e$. Find a spanning tree $T$ that minimizes $(\sum_{e \in T} A_e) \times (\sum_{e \in T} B_e)$. You algorithm should run in polynomial time.

**Problem 9:** (Second MST) Assume the edge costs are all distinct. Design a polynomial time algorithm that finds a second-best minimum spanning tree (i.e., a tree with the minimum cost among all spanning trees except the MST).