CS 473: Algorithms, Fall 2010
HW 6 (due Tuesday, October 19)

This homework contains four problems. **Read the instructions for submitting homework on the course webpage**. In particular, *make sure* that you write the solutions for the problems on separate sheets of paper; the sheets for each problem should be stapled together. Write your name and netid on each sheet.

Collaboration Policy: For this home work, Problems 1-3 can be worked in groups of up to 3 students each.

Problem 0 should be answered in Compass as part of the assessment HW6-Online and should be done individually.

0. (10 pts) HW6-Online on Compass.

1. (25 pts) A party of n people have come to dine at a fancy restaurant and each person has ordered a different item from the menu. Let $D_1, D_2, \ldots, D_n$ be the items ordered by the diners. Since this is a fancy place, each item is prepared in a two-stage process. First, the head chef (there is only one head chef) spends a few minutes on each item to take care of the essential aspects and then hands it over to one of the many sous-chefs to finish off. Assume that there are essentially an unlimited number of sous-chefs who can work in parallel on the items once the head chef is done. Each item D_i takes h_i units of time for the head chef followed by s_i units of time for the sous-chef (the sous-chefs are all identical). The diners want all their items to be served at the same time which means that the last item to be finished defines the time when they can be served. The goal of the restaurant is to serve the diners as early as possible. Give an efficient algorithm to decide the order in which the head chef should prepare the items so as to minimize the time to serve the diners.

2. (30 pts) A sequence is *palindromic* if it is the same whether read left to right or right to left. An example is $m, a, l, a, y, a, l, a, m$ (*Malyalam* is a Southern Indian language). Given a sequence $a_1, a_2, \ldots, a_n$ give an $O(n^2)$ algorithm to compute a *longest* palindromic *subsequence* of the given sequence. For example, the sequence below

   $$A, C, G, T, G, T, C, A, A, A, A, T, C, G$$

   has many palindromic subsequences, include A, C, G, C, A and A, A, A, A.

   Extra credit (15 pts): Give an algorithm for the above problem that uses only $O(n)$ space. No formal proof needed but algorithm has to be clear and understandable; add explanation as necessary.

3. (35 pts) You are given an undirected graph $G = (V, E)$ with non-negative edge weights: $w(e)$ denotes the weight of edge e. Let m and n be the number of edges and nodes in G respectively. Suppose you are told that the graph has at most k *distinct* edge weights.

   - Describe an algorithm that first checks whether there are only k distinct edge weights or not in $O(m \log k)$ time.

- Describe an algorithm that finds the MST of G in $O((m+n)\log k)$ time. *Hint:* Use Prim's algorithm with appropriate data structure modifications.

Partial credit of 20 pts for an algorithm that runs in $O((m+n)\cdot k)$ time for both of the above steps.