15-451 Homework 4

March 17, 2008

Due: Mon-Fri, March 24-28

1. This is an oral presentation assignment. You should work in groups of three. At some point before **Sunday, March 23 at 11:59pm** your group should sign up for a 1-hour time slot on the signup sheet on the course web page.
2. Each person in the group must be able to present every problem. The TA/Professor will select who presents which problem. The other group members may assist the presenter.
3. Your are not required to hand anything in at your presentation, but you may if you choose.
4. Direct questions to Bryant Lee (bryantl@cs.cmu.edu)

1 Question 1

Boruvka's MST Algorithm. Boruvka's MST algorithm (from 1926) is a bit like a distributed version of Kruskal. We begin by having each vertex mark the shortest edge incident to it. (For instance, if the graph were a 4-cycle with edges of lengths 1, 3, 2, and 4 around the cycle, then two vertices would mark the "1" edge and the other two vertices will mark the "2" edge.) For the sake of simplicity, assume that all edge lengths are distinct so we don't have to worry about how to resolve ties. This creates a forest F of marked edges. (Convince yourself why there won't be any cycles!) In the next step, each tree in F marks the shortest edge incident to it (the shortest edge having one endpoint in the tree and one endpoint not in the tree), creating a new forest F'. This process repeats until we have only one tree.

1. Show correctness of this algorithm by arguing that the set of edges in the current forest is always contained in the MST.
2. Show how you can run each iteration of the algorithm in $O(m)$ time with just a couple runs of Depth-First-Search and no fancy data structures (heaps, union-find). Remember, this algorithm was from 1926!
3. Prove an upper bound of $O(mlogn)$ on the running time of this algorithm.

2 Question 2

Road trip! Velma and the gang are going on a road trip to San Francisco immediately after their midterms. To plan the trip, they have laid out a map of the U.S., and marked all the places they think might be interesting to visit along the way. However, the requirements are:

1. Each stop on the trip must be closer to SF than the previous stop.

2. The total length of the trip can be no longer than D. Velma wants to visit the most places subject to these conditions. As a first step, she creates a DAG with n nodes (one for each location of interest) and an edge from i to j if there is a road from i to j and j is closer to SF than i. Let d_{ij} be the length of edge (i, j) in this graph.

   Help out Velma by giving an $O(mn)$-time algorithm to solve her problem. Specifically, given a directed acyclic graph (DAG) G with lengths on the edges, a start node s, a destination node t, and a distance bound D, your algorithm should find the path in G from s to t that visits the most intermediate nodes, subject to having total length $\leq D$. (Note that in general graphs, this problem is NP-complete: in particular, a solution to this problem would allow one to solve the traveling salesman problem. However, the case that G is a DAG is much easier.)

3 Question 3

You are given coins of denominations $a_1, a_2, ..., a_n$ (in cents), and are required to make change for a certain amount x (also in cents). In all of the below questions, your job is to design an efficient dynamic programming algorithm. In each case, you should give either a (top-down) recursive procedure for the DP, or a bottom-up array-filling procedure. Explain what each entry of the array stands for, and give the algorithm's running time. A proof of correctness is not required.

1. Give an algorithm for determining whether you can make change for x using the given denominations.
2. Give an algorithm for determining whether you can make change for x using at most one coin of each given denomination.
3. Give an algorithm for determining whether you can make change for x using at most k coins in total from among the given denominations.