15-451 Algorithms, Fall 2011

Homework # 4 **Due: Tue-Thu, October 25-27, 2011**

Ground rules:

- This is an oral presentation assignment. You should work in groups of three. At some point before **Saturday, October 22 at 11:59pm** your group should sign up for a 1-hour time slot on the signup sheet on the course web page.
- 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.
- You are not required to hand anything in at your presentation, but you may if you choose.

Problems:

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.

   (a) Show correctness of this algorithm by arguing that the set of edges in the current forest is always contained in the MST.

   (b) 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!

   (c) Prove an upper bound of $O(m \log n)$ on the running time of this algorithm.

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:

   (a) Each stop on the trip must be closer to SF than the previous stop.

   (b) 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. **Dynamic programming for (fun and) profit.** An *option* (specifically, an "American call option") gives the holder the right to purchase some underlying asset (e.g., one share of IBM) at some specified exercise price (e.g., \$200) within some specified time period (e.g., 1 year). For instance, if the price of IBM went up to \$212 in this period, you could exercise the option and then sell the stock, making \$12 (or you could keep the option, hoping the price will increase further before the option expires).

   If you have a probabilistic model for how a stock will behave, then this gives you a well-defined notion of the *value* of a given option: the expected profit for having the option if you were to follow the optimal strategy for deciding when to exercise it under that model.

   For example, to take an easy case, suppose we have an option to buy one share of IBM at \$200 that expires right now. If IBM is currently going for \$210, then the value of this option is \$10. If IBM is currently going for \$190, then the value of this option is \$0 (we wouldn't want to exercise it). However, suppose the option expires tomorrow, and suppose our model says that each day, with probability 1/4 the share price goes up by \$20, and with probability 3/4 the share price goes down by \$10 . In that case, if IBM is currently worth \$190, then the value of this option is \$2.50 because that is our expected gain if we use the optimal strategy "wait until tomorrow, and then exercise the option if IBM went up" (our expected gain under this strategy would be $\frac{1}{4} \times 10 + \frac{3}{4} \times 0$). If IBM is currently worth \$210, then the value of the option is \$10 (because if you work it out, our optimal strategy in this case is to exercise the option right away).

   Formally, the value of an option is the expected profit it will produce under the optimal strategy for using the option, given our probabilistic model for the stock. Note that the optimal strategy need not commit in advance to what day the option will be exercised: the date it gets exercised (if ever) may depend on how the stock has performed so far.[1]

   Assume stock prices are *integers* between 0 and B. Suppose we are given a probabilistic model p_{ij} for how the stock behaves: specifically, if the stock has price i on day t, then p_{ij} is the probability that the stock will have price j on day $t+1$. So, for each i, $\sum_j p_{ij} = 1$.

   (a) Give a dynamic-programming algorithm to calculate the value of an option of exercise price X that expires T days in the future, given that the current price of the stock is S. The running time of your algorithm should be $O(B^2T)$. Hint: think backwards.

   (b) Suppose our option can only be exercised at exactly time T, rather than any time $\leq T$ (this is called a "European option"). Describe an algorithm to solve for the option's value that runs in time $O(B^3 \log T)$. Hint: think matrices.

[1] For example, suppose your probabilistic model is that the stock performs a random walk in the range [\$170, \$220]. That is, each day, the stock has a 50/50 chance of going up or down by \$1, unless it is at one of the "barriers" in which case it would stay put rather than exit the range. In this case, you would always want to exercise the option if it ever reaches \$220.