CSE 331: Introduction to Algorithm Analysis and Design Fall 2011

HOMEWORK 7

Due Friday, November 4, 2011 by 1:15pm in class

Please submit each problem separately, i.e. each problem should begin on a new page and only the pages for one problem should be stapled together. Failure to do so might result in some problem(s) not being graded.

For general homework policies and our suggestions, please see the policy document.

No collaboration is allowed on the first problem.

---

1. (40 points) (**Your must work on this problem on your own: NO collaboration is allowed**)

   In class I mentioned that Dijkstra's algorithm does not work when the edge lengths can be negative. Show this is possible in the strongest possible sense by giving an example that exhibits this. In particular, come up with an example of a directed graph (with **no** directed cycles) along with edge lengths (**exactly** one of which is negative) and two nodes s and t in the graph such that when Dijkstra's algorithm is run on the graph, it does not output the shortest path between s and t. (As before, the length of a path is the sum of the lengths of the edges in it.)

   Note: You have the full freedom in picking the graph.

2. (45 points) Mr. GluttonTraveller wants to drive from Buffalo to Seattle. However, all he cares about is how much he has to pay for Hamburgers during his trip. Towards this end, he secures a special "Hamburger map" for the US which shows Hamburger joints along with the following information: (i) Cost of a Hamburger at each joint and (ii) shows two Hamburger joints to be connected if there is a path between them– you should assume that these "edges" are directed. (You can assume that both Buffalo and Seattle have a Hamburger joint.) Finally, if Mr. GluttonTraveller is at a place that has a Hamburger joint, then he always buys a Hamburger (and of course eats it too). You can assume that there are n Hamburger joints and there are m "edges."

   Mr. GluttonTraveller missed the class on greedy algorithms. Help him by designing an algorithm that computes the driving path from Buffalo to Seattle that will result in him spending the minimum amount of money on Hamburgers.

   Prove the correctness of your algorithm and analyze the running time of your algorithm. You can assume that the map is represented as an adjacency list.

   (*Hint:* It *might* be easier to modify the input first.)

3. (15 points) Design an $O(m+n)$ time algorithm for the following problem. Given a Directed Acyclic Graph (or DAG) G with arbitrary weights on the edges (i.e. they can be positive or negative) with n vertices and m edges, and given two vertices s and t, compute the shortest path from s to t in G. (You can assume that G is given to you in the adjacency list format.)

   (*Note:* In case you missed it, note that the weights can be negative and the run time of the algorithm needs to be a linear in the input size.)