CSE 331: Introduction to Algorithm Analysis and Design Fall 2010

HOMEWORK 10
Due Friday, December 10, 2010 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.

Do not turn in Q 2.

1. $(10 + 30 = 40$ points) In class, we will see a dynamic program that computes the shortest path even some of the edge lengths are negative (but there are no negative cycle). Earlier in the semester, I had mentioned that Dijkstra's algorithm does not work when the edge lengths can be negative.

   (a) Give an example of a directed graph along with edge lengths (*exactly* one of which is negative while the rest of the edge weights are all positive) 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.

   (b) Argue whether the following is true or false: There is an input graph which is a tree and has at least one non-negative edge (but no negative cycle) such that Dijkstra's algorithm fails to output the shortest (simple) path between some pair of vertices.

2. (**Do NOT turn this part in**) Exercise 1 in Chapter 6.

   Note: If you get confused by the language in part (c) that talks about running time being independent of the weight values recall that the running time for the weighted interval scheduling algorithm we saw in class also has running time independent of the magnitude of the values.

   Hint: For part (c), don't directly try to apply the weighted interval scheduling problem. Instead use the same design principle as we used to come up with the dynamic program for the weighted interval scheduling problem: i.e., come up with a recursive definition and then observe that only polynomially many values need to be computed and stored. (This hint is also valid for part (b) in the next question.)

3. $(20 + 25 = 45$ points) Exercise 2 in Chapter 6.

   Hint: For part (b) convince yourself that one should always schedule a job on the last day and then use it. (BTW in your homework, you will also have to convince Alex/Jeff why you are convinced if you choose to use the hint, i.e. just using the hint as given will result in loss of points.)

4. (15 points) In the divide and conquer algorithm for finding the closest two points, we saw that if two points in S were $< \delta$ apart then they could be at most 15 positions apart in S_y.[1]

---

[1]If you do not remember what I am talking about, see (5.10) in the book.

Come up with a number $\alpha < 15$ and prove that the result above still holds if one replaces 15 with α

Note: To get *any* credit your proof must work for some $\alpha \le 12$. To get full credit, your proof must work for some $\alpha \le 10$.

Hint: It might be useful to define concretely what happens when a point is *on* one of the lines at $x = x^*$, $x = x^* - \delta$ and $x = x^* + \delta$.