Homework 4 – Due Friday, September 24, 2010

Please refer to the general information handout for the full homework policy and options.

Reminders

- Your solutions are due before the lecture. Late homework will not be accepted.

- Collaboration is permitted, but you must write the solutions by yourself without assistance, and be ready to explain them orally to a member of the course staff if asked. You must also identify your collaborators. Getting solutions from outside sources such as the Web or students not enrolled in the class is strictly forbidden.

- To facilitate grading, please write down your solution to each problem on a separate sheet of paper. Make sure to include all identifying information and your collaborators on each sheet. Your solutions to different problems will be graded separately, possibly by different people, and returned to you independently of each other.

- For problems that require you to provide an algorithm, you must give a precise description of the algorithm, together with a proof of correctness and an analysis of its running time. You may use algorithms from class as subroutines. You may also use any facts that we proved in class or from the book.

Reading  Review chapters 4 and 5.1–5.3 in Kleinberg Tardos.

Problems to be handed in

1. (Shortest paths for other measures of distance) Dijkstra’s algorithm assumes that the cost of a path is the sum of the lengths of the edges in the graph. However, path costs are not always like that. For example,

   (a) In a road network, the travel time on a path might have to include extra time for turning off a busy road. Thus, the cost of path $P$ might be $\sum_{e \in P} \ell(e) + c_1(\# \text{ of left turns}) + c_2(\# \text{ of right turns})$, where $c_1$ and $c_2$ represent the extra time needed for each kind of turn.

   (b) If the edges are links in a computer network and edge weights are bandwidths, then the “value” of a path would be the minimum of the bandwidths along the path. Note that in this case we would want to maximize the value, not minimize it.

   (c) Even if the path cost is still a simple sum, the edge weights may change over time. For example, in the “Canadian travel” problem (KT, Chapter 4, problem 18), the time of travel along an edge $(u, v)$ is a function of the time $t$ at which you depart from $u$. See the problem statement in the book for details.

For each of these three types of “distances”, show how to modify Dijkstra’s algorithm so that it computes the best path according to the new distance measure. In each case, indicate
• which steps of the algorithm you would change,
• how these changes affect the running time, and
• what changes are necessary to proof of correctness we saw in class to conclude that the
modified algorithm is correct.

For part (1a) above, assume that given the edges (=streets) by which a path enters and leaves
a given vertex (=intersection), there is a constant-time procedure that determines if the driver
needed to turn left, turn right, or go straight.

[Note: After completing this problem, do problems 19 and 20 in chapter 4 as exercises; do not
hand them in.]

2. KT, Problem 31 (light subgraphs with short paths).

3. (Recursion-Tree Method) Use the recursion-tree method to solve the following recurrences.
Express your answer using $\Theta$-notation. Note: It may help to review sums of geometric and
arithmetic progressions (e.g. sums like $\sum_{i=1}^{n} r_i$ or $\sum_{i=1}^{n} i$.)

(a) $T(n) = 7T(n/2) + cn$, where $c \geq 0$ is a constant.
(b) $T(n) = T(n-2) + c \log(n)$, where $c \geq 0$ is a constant.
(c) $T(n) = \sqrt{n}T(\sqrt{n}) + cn$, where $c \geq 0$ is a constant.

In all three cases, you may assume $T(n)$ is constant for $n < 10$. 

2