Homework 3 – Due Friday, September 17, 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 chapter 4.1-4.5 in Kleinberg Tardos.

Exercises  These should not be handed in, but the material they cover may appear on exams:

1. Exercises in Chapter 4.

2. Point out the error in the following proof by induction.

Claim 1  In any set of $h$ horses, all horses are the same color.

Proof: We proceed by induction on the number $h$ of horses. (Base case) If $h = 1$, then there is only one horse in the set, and so all the horses in the set are clearly the same color.

(Induction Step) For $k \geq 1$, we assume that the claim holds for $h = k$ and prove that it is true for $h = k + 1$. Take any set $H$ of $k + 1$ horses. We show that all horses in this set are of the same color. Remove one horse from this set to obtain the set $H_1$ with just $k$ horses. By the induction hypothesis, all the horses in $H_1$ are the same color. Now replace the removed horse and remove a different one to obtain a the set $H_2$. By the same argument, all the horses in $H_2$ are the same color. Therefore all the horses in $H$ must be the same color, and the proof is complete.

3. (Analysis of $d$-ary heaps) A $d$-ary heap is like a binary heap, described in Chapter 2.5 of Kleinberg Tardos, with the exception that non-leaf nodes have $d$ children instead of 2.
(a) How would you represent a $d$-ary heap in an array?

(b) Implement $\text{Parent}(i)$ that, given the index $i$ of a node, returns the index of its parent and $\text{Child}(i, k)$ that, given the index $i$ of a node, returns the index of its $k$th child.

(c) What are the minimum and the maximum number of elements in a $d$-ary heap of height $h$?

(d) Design an efficient implementation of $\text{Heapify-Up}$ in a $d$-ary min-heap, analogous to the procedure on page 61 of KT. Analyze the running time of your algorithm in terms of $d$ and $n$.

(e) Design an efficient implementation of $\text{Heapify-Down}$ in a $d$-ary min-heap, analogous to the procedure on page 63 of KT. Analyze the running time of your algorithm in terms of $d$ and $n$.

(f) Suppose we implement a priority queue using a $d$-ary heap. Give the running times of all operations, described on pages 64–65 of KT, in terms of $d$ and $n$.

4. (MST with negative-weight edges) Give a $O(m \log n)$ algorithm which takes as input a connected, undirected graph $G = (V, E)$ and distinct weights $w$, some of which may be negative, and outputs a minimum spanning subgraph $F \subseteq E$, i.e. it outputs a set of edges $F$ such that $(V, F)$ is connected and $\sum_{e \in F} w_e$ is minimized. Note that $F$ need not be a tree.

5. Suppose $G$ is a connected, undirected graph in which all edge weights $w_e$ are positive and distinct. Consider the graph $G'$ with the same vertices and edges as $G$, but with edge weights $w'_e = w_e^2 + 6$. Prove or disprove: For every pair of vertices $u$ and $v$, the shortest path from $u$ to $v$ is the same in $G$ as in $G'$.

6. Suppose $G$ and $G'$ are as above. Prove or disprove: The minimum spanning tree of $G'$ is the same as the minimum spanning tree of $G$.

Problems to be handed in

Page limits: The answer to each problem should fit in 2 pages (or one double-sided sheet) of paper. Longer answers will be penalized.
3. [Extra credit, can be handed in on Wednesday, September 22.] For any edge $e$ in any graph $G$, let $G \setminus e$ denote the graph obtained by deleting $e$ from $G$.

(a) Suppose we are given a directed graph $G$ in which the shortest path from vertex $s$ to vertex $t$ passes through every vertex of $G$. Describe an algorithm to compute the shortest-path distance from $s$ to $t$ in $G \setminus e$, for every edge $e$ of $G$, in $O(E \log V)$ time. Your algorithm should output a set of $E$ shortest-path distances, one for each edge of the input graph. You may assume that all edge weights are non-negative. [Hint: If we delete an edge of the original shortest path, how do the old and new shortest paths overlap?]

(b) Let $s$ and $t$ be arbitrary vertices in an arbitrary directed graph $G$. Describe an algorithm to compute the shortest-path distance from $s$ to $t$ in $G \setminus e$, for every edge $e$ of $G$, in $O(E \log V)$ time. Again, you may assume that all edge weights are non-negative.