Problem Set 2

CS157 - Spring 2009

Due: Tuesday February 24, 2008 10:30am

Problem 1: Hadamard Matrices

The *Hadamard matrices* $H_0, H_1, H_2, \ldots$ are defined as follows:

- H_0 is the 1×1 matrix $[1]$.
- For $k > 0$, H_k is the $2^k \times 2^k$ matrix

$$H_k = \left[\begin{array}{c|c} H_{k-1} & H_{k-1} \\ \hline H_{k-1} & -H_{k-1} \end{array}\right]$$

Throughout this problem assume that the numbers involved are small enough that basic arithmetic operations like addition and multiplication take unit time.

(a) Display the Hadamard matrices H_0, H_1 and H_2.

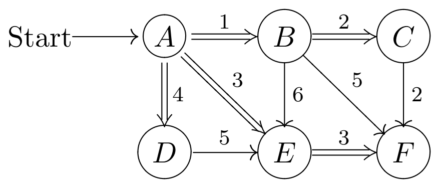
(b) Compute $H_0 \cdot (z)$, $H_1 \cdot \begin{bmatrix} x \\ y \end{bmatrix}$ and $H_2 \cdot \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$, where a, b, c, d, x, y and z are numbers.

(c) Assume you have a black-box that computes $H_{k-1} \cdot v$ for any column vector v (of the appropriate size). Show how to use two calls to this black box, plus $O(2^k)$ additional work, to compute $H_k \cdot u$ for any column vector u (of the appropriate size).

(d) Give a recursive algorithm that computes $H_k \cdot v$. Analyze its runtime. Express the runtime in terms of the size of the input vector $n = 2^k$.

Problem 2: Shortest path verification

(a) Do the doubled edges in the following graph form a shortest-path tree (rooted at A)? Justify your answer.



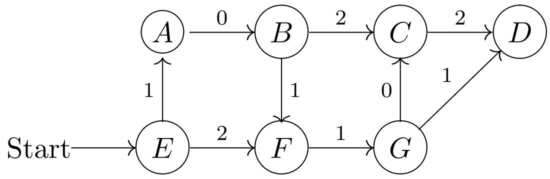
(b) Generalizing the above, you are given a directed graph $G = (V, E)$ with weighted edges, along with a specific node $s \in V$ and a tree $T = (V, E')$, $E' \subseteq E$. Give an efficient algorithm that decides whether or not T is a shortest-path tree for G with starting point s.

(c) Argue briefly that your algorithm runs in linear time.

(d) Prove that your algorithm is correct.

Problem 3: Dijkstra's algorithm with integer weights

(a) Show how Dijkstra's algorithm can be made to run in $O(|V| + |E|)$ time if the input graph has edge weights that are only $0, 1, 2$. The following example graph may be helpful:



(b) Show how Dijkstra's algorithm can be made to run in $O(W|V| + |E|)$ time if the input graph has edge weights that are integers in the range $0, 1, 2, \ldots, W$.

(c) Describe an alternative implementation with runtime $O((|V| + |E|) \log W)$.

Problem 4: Minimum spanning trees

The following statements about minimum spanning trees may or may not be correct. Indicate whether or not each statement is correct and give a counterexample to *every* incorrect statement. Give a short proof of *one* of the true statements of your choice. Always assume that the graph $G = (V, E)$ is undirected and connected. Do not assume that edge weights are distinct unless this is specifically stated.

(a) If graph G has more than $|V| - 1$ edges, and there is a unique heaviest edge, then this edge cannot be part of a minimum spanning tree.

(b) If G has a cycle with a unique heaviest edge e, then e cannot be part of any MST.

(c) Let e be any edge of minimum weight in G. Then e must be part of some MST.

(d) If the lightest edge in a graph is unique, then it must be part of every MST.

(e) If e is part of some MST of G, then it must be a lightest edge across some cut of G.

(f) If G has a cycle with a unique lightest edge e, then e must be part of every MST.

(g) The shortest-path tree computed by Dijkstra's algorithm is necessarily an MST.

(h) The shortest path between two nodes is necessarily part of some MST

(i) For any $r > 0$, define an *r-path* to be a path whose edges all have weight $< r$. If G contains an r-path from node s to t, then every MST of G must also contain an r-path from node s to node t.