Feb 26

(1) Describe an efficient algorithm that given an undirected graph $G$, determines a spanning tree of $G$ whose largest edge weight is minimum, over all spanning trees of $G$. Give an argument justifying your algorithm.

(2) The diameter of a tree $T = (V,E)$ is given by

$$\max_{u,v \in V} \delta(u,v)$$

where $\delta(u,v)$ is the distance between $u$ and $v$ in the tree $T$. Give an $O(|V|)$ algorithm for computing the diameter of the tree. Write a proof of correctness for your algorithm.

(3) In a directed graph, a **get-stuck** vertex is one that has in-degree $|V| - 1$ and out-degree 0. Assume that the adjacency matrix representation is used. Design an $O(|V|)$ algorithm to determine if a given graph has a get-stuck vertex. (Yes, this problem can be solved without even looking at the entire input matrix.) Write a proof of correctness for your algorithm.

(4) Assume that we have a network (a connected undirected graph) in which each edge $e_i$ has an associated bandwidth $b_i$. If we have a path $P$, from $s$ to $v$, then the capacity of the path is defined to be the minimum bandwidth of all the edges that belong to the path $P$. We define $\text{capacity}(s,v) = \max_{P(s,v)} \text{capacity}(P)$. (Essentially, $\text{capacity}(s,v)$ is equal to the maximum capacity path from $s$ to $v$.) Give an efficient algorithm to compute $\text{capacity}(s,v)$, for each vertex $v$; where $s$ is some fixed source vertex. Show that your algorithm is “correct”, and analyze its running time. (Design something that is no more than $O(|V|^2)$, and with the right data structures takes $O(|E| \log |V|)$ time.)

(5) Let $G$ be a directed graph. The vertices of $G$ have been numbered 1…$n$ (where $n$ is the number of vertices in $G$). Let $\text{small}(i) = \min\{j|j$ is reachable from $i\}$. In other words, for a vertex numbered $i$, $\text{small}(i)$ is the smallest numbered vertex reachable from it. Design an $O(V + E)$ algorithm to compute $\text{small}(i)$ for all vertices in the graph.