Introduction to Analysis of Algorithms	Homework 1 Supplement
CS4820 Spring 2013	Tuesday, January 22, 2013

For Exercise 2, it will help to recall the relationship between the graph-theoretic concepts of acyclicity, strong connectivity, and topological sort (which you presumably learned about in CS 2110/3110) and the concepts of preorder, partial order, and total order from discrete math (CS 2800).

(i) A *preorder* $\leq$ is a reflexive and transitive binary relation. *Reflexive* means $x \leq x$ for all x. *Transitive* means for all x, y, z, if $x \leq y$ and $y \leq z$, then $x \leq z$.

(ii) A *partial order* is a preorder that satisfies the additional condition of *antisymmetry*: for all x, y, if $x \leq y$ and $y \leq x$, then $x = y$.

(iii) A *total order* (or *linear order*) is a partial order that satisfies the additional condition of *totality*: for all x, y, either $x \leq y$ or $y \leq x$.

As mentioned in the statement of the problem, an arbitrary finite directed graph $G = (V, E)$ determines a preorder $\leq$ on its nodes in which $x \leq y$ if there is a directed E-path of length 0 or greater from x to y. The *length* of a path is the number of edges in the path. Note that $\leq$ is reflexive, since there is a path of length 0 from every node to itself, and transitive, since if there is a path from x to y and a path from y to z, then there is a path from x to z. Every preorder $\leq$ on a finite set V can be represented by a graph in this way.

(iv) A *cycle* in G is a path with the same start and end point. Every node is the start and end point of a trivial cycle of length 0; a directed graph is called *acyclic* if it has no other cycles besides these. A directed acyclic graph is called a *dag*.

(v) If G is acyclic, it is always possible to order the nodes as $x_1, \ldots, x_n$ so that every edge goes from a lower-numbered node to a higher-numbered node; that is, if $(x_i, x_j) \in E$, then $i < j$. Such an ordering is called a *topological sort* of G. One can topologically sort a given dag in linear time (see K&T §3.6).

(vi) Define an equivalence relation $\equiv$ on nodes as follows: $x \equiv y$ if there is a directed path from x to y and a directed path from y to x. Equivalently, $x \equiv y$ if there is a cycle containing both x and y. One can show that $\equiv$ is an equivalence relation (reflexive, symmetric, and transitive), therefore partitions V into a set of nonempty disjoint equivalence classes whose union is V. A *strongly connected component* (or just *strong component*) of G is an equivalence class of $\equiv$. One can find all the strong components in linear time (see K&T §3.5).

There is a close relationship between the discrete math concepts (i)–(iii) on finite sets and the graph-theoretic concepts (iv)–(vi). As mentioned, the relation $\leq$ defined by paths in G is always a preorder. It is a partial order iff[1] G is acyclic. If G is acyclic and we topologically sort the nodes to get the numbering $x_1, \ldots, x_n$, then add the edges (x_i, x_{i+1}) for $1 \leq i \leq n-1$ to get a new graph G', then the order $\leq'$ represented by G' is a total order extending the partial order $\leq$ represented by G. By *extending* we mean that for all nodes x, y, if $x \leq y$, then $x \leq' y$.

Now comes a really interesting construction. Even if G has nontrivial cycles, we can squeeze it into a dag by collapsing the strong components into single nodes. The resulting collapsed graph is called the *quotient graph* (with respect to $\equiv$), and it is always acyclic. Formally, let $[x] = \{y \mid x \equiv y\}$. This is the $\equiv$-equivalence class of x, that is, the unique strong component of G containing x. The quotient graph is $G/{\equiv} \; = \; (V/{\equiv}, \, E/{\equiv})$, where

- $V/{\equiv} \; = \; \{[x] \mid x \in V\}$, the set of all strong components,
- $E/{\equiv} \; = \; \{([x], [y]) \mid (x, y) \in E, \; x \not\equiv y\}$.

One must prove formally that $G/{\equiv}$ is acyclic, but this is not difficult. Intuitively, every cycle in G lives inside a single strong component, so it gets collapsed into a single node. Moreover, there is a directed path from x to y in G iff there is a directed path from $[x]$ to $[y]$ in $G/{\equiv}$.

[1] iff = if and only if