Assignment 4

Post Date: 10 Nov 2010  Due Date: 17 Nov 2010, 14:30
You are permitted and encouraged to work in groups of two.

Problem 1: Sequence of Operations  
Consider sequences of operations MAKESET, FIND with path compression, and weighted UNION where all UNION operations are performed before the first FIND operation.

(a) Show that the amortized cost for \( n \) operations is in \( \mathcal{O}(n) \).

(b) Does (a) hold if FIND is still with path compression but UNION is unweighted?

(c) Does (a) hold if UNION is still weighted but FIND is without path compression?

Problem 2: Union-Find with Path Compression

(a) Give a pseudocode for FIND with path compression similar to the pseudocode of FIND without path compression from the lecture.

(b) Consider FIND with the following alternative path compression: After traversing the path from a vertex to its root, we update the parent pointer of each vertex along the path to point to its grandparent. Consider, e.g., subpath

\[ i \rightarrow j \rightarrow k \rightarrow l \rightarrow \cdots \]

Performing FIND\((i)\) with alternative path compression results in \( k \) being predecessor of \( i \) and \( l \) being predecessor of \( j \). Direct successors of the root keep the root as predecessor.

Go through the proof of the Theorem of Hopcroft \& Ullman and find the inferences that require FIND to be implemented with path compression. Is the proof still correct if the alternative path compression is used?

[please turn over]
Problem 3: Least Common Ancestor 8 Points

Let $T = (V, E)$ be a directed tree with root $r \in V$ and let $P \subseteq \{\{u, v\} | u, v \in V\}$ be a set of unordered pairs of vertices. For each $v \in V$ let $\Pi_v = \{r, \ldots, v\} \subseteq V$ denote the sequence of vertices along the path from $r$ to $v$ and let $d(v) = |\Pi_v| - 1$ denote the depth of $v$ in $T$.

The least common ancestor of pair $\{u, v\} \in P$ is defined as $\bar{w} \in V$ with $\bar{w} \in \Pi_v \cap \Pi_u$ and $d(\bar{w}) > d(w)$ for all $w \in \Pi_v \cap \Pi_u$.

LCA($r$) traverses $T$ to determine the least common ancestors of all pairs $\{u, v\} \in P$. At the beginning, all vertices are unmarked.

**Algorithm 1: LCA($u$)**

1. MAKESET($u$)
2. ancestor[Find($u$)] ← $u$
3. foreach child $v$ of $u$ in $T$ do
   4. LCA($v$)
   5. UNION(Find($u$), Find($v$))
   6. ancestor[Find($u$)] ← $u$
7. mark $u$
8. foreach $v$ with $\{u, v\} \in P$ do
   9. if $v$ is marked then
      10. print 'Least common ancestor of' $u$ 'and' $v$ 'is' ancestor[Find($v$)]

(a) Show that, when Line 7 in LCA($u$) is executed, the set Find($u$) contains all vertices of the subtree $T_u \subseteq T$ with root $u$.

**Hint:** Do an induction on the height $h(u)$ of $u$. ($h(u) = 0$ if $u$ is a leaf)

(b) Show that the number of sets in the UNION-FIND data structure at the time of call LCA($v$) equals $d(v)$.

**Hint:** The recursive calls of LCA specify a traversing order of $T$ which implies an order on $V$. Do an induction on the position of $v$ in this order.

(c) Prove that LCA($r$) determines the least common ancestors of all $\{u, v\} \in P$ correctly.

**Hint:** Differentiate the two cases: 1. w.l.o.g. $v \in T_u$ and 2. $v \notin T_u$ and $u \notin T_v$

(d) Analyze the running time of LCA($r$).