Depth Determination Problem

Andrzej

February 1, 2013

(a) If the trees are represented, as suggested, by their isomorphic copies where each node has a parent pointer (call them "up-trees"), then one can grow a linear tree (with one leaf) by repeatedly grafting single vertex tree on the leaf of a smaller linear tree (starting with the trivial tree). Determining the depth of the leaf after every graft will result in the total complexity of $\Theta(m^2)$. [1 point]

(b) The problem suggests that a rooted tree $T$ be represented by "an up-tree" $S$, where each node’s depth in $T$, say $k = d_T(x)$, is equal to the sum of "pseudo-depth" values $pd$ of the nodes on the FindPath from $x$ to the root $q$ of $S$. Note that $q$ does not have to be the root of $T$ and therefore $pd(q)$ has the value $d_T(q)$. If the node $x$ in $T$ happens to be a child of $q$ in $S$ (not necessarily a child in $T$!), then – by the definition – its pseudo-distance $pd(x) = k - pd(q)$.
[1 point for MAKE-TREE(v) implemented by $pd(v) = 0$ and $p(v) = v$]

(c) If the Find operation (of $y$ in $T$ represented by $S$) involves path compression, then upon finding the root $q$, not only the parent pointers of the nodes on the path have to be changed to $q$, but also their pseudo-distances need to be updated. This can be achieved by traversing the path in the opposite ("downward") direction while computing the true depth values.
[2 points for FIND-DEPTH]

(d) To correctly represent the tree GRAFT($r$, $v$), your algorithm needs to perform the Union (by rank) of the two trees $S_1$=Find($r$) and $S_2$=Find($v$) making sure that the pseudo-distances are updated correctly. Note that every node formerly in the tree $T_1$ rooted at $r$ will have the (true) depth increased by $1 + d_{T_2}(v)$, while the depth of every node formerly in $T_2$ will remain the same. Therefore, depending of which root (of $S_1$ or of $S_2$) will acquire a new parent as the result of Union, their pseudo-distances will have to be updated accordingly: the new (overall) root will have the pseudo-distance equal to its (true) depth in the tree GRAFT($r$, $v$) and its new child will have the pseudo-distance decreased by that amount (and possibly increased by $1 + d_{T_2}(v)$).
[3 points for GRAFT]
(e) This implementation uses the augmented Union and Find operations and therefore has the amortized complexity of $O(\alpha(m))$ [1 point]