Assignment 4

Post Date: 14 Nov 2012  Due Date: 21 Nov 2012, 14:30
You are permitted and encouraged to work in groups of two.

Problem 1: Sequence of Operations 6 Points

Consider sequences of operations \textsc{makeset}, \textsc{find} with path compression, and weighted \textsc{union} where all \textsc{union} operations are performed before the first \textsc{find} operation.

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

(b) Does (a) hold if \textsc{find} is still with path compression but \textsc{union} is unweighted?

(c) Does (a) hold if \textsc{union} is still weighted but \textsc{find} is without path compression?

Problem 2: Union-Find with Path Compression 4 Points

(a) Give a pseudocode for \textsc{find} with path compression similar to the pseudocode of \textsc{find} without path compression from the lecture.

(b) Consider \textsc{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 \textsc{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 \textit{Theorem of Hopcroft \\& Ullman} and find the inferences that require \textsc{find} to be implemented with path compression. Is the proof still correct if the alternative path compression is used?
Problem 3: Equivalence of Finite Automata

Let $A_1$ and $A_2$ be finite automata with sets of states $Q_1$ and $Q_2$, respectively.

Determine for each state $p \in Q_1$ the set $Q_p := \{q \in Q_2; q \equiv p\}$. Decide whether $A_1$ and $A_2$ are equivalent.

Problem 4: Prüfer Sequence

(a) Determine for the following tree the Prüfer sequence.

(b) Construct from the Prüfer sequence $(4, 9, 4, 6, 6, 5, 1)$ the corresponding tree.