University of Konstanz
Department of Computer & Information Science
Prof. Dr. Ulrik Brandes / Dr. Sven Kosub / Dr. Sabine Cornelsen and Natalie Indlekofer

Design and Analysis of Algorithms
WS 2012/2013

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 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 — 4 Points

(a) Give a pseudocode for Find with path compression similar to the pseudcode 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

$$i \to j \to k \to l \to \cdots$$

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.

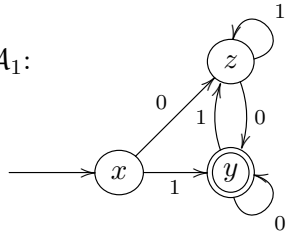
Go through the proof of the *Theorem of Hopcroft & Ullman* and find the inferences that require Find to be implemented with path compression. Is the proof still correct if the alternative path compression is used?

[please turn over]

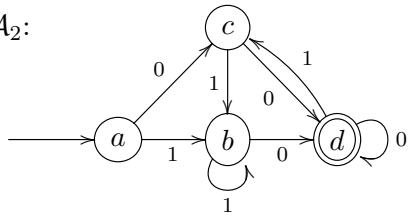
Problem 3: Equivalence of Finite Automata **6 Points**

Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be finite automata with sets of states Q_1 and Q_2, respectively.

$\mathcal{A}_1$:



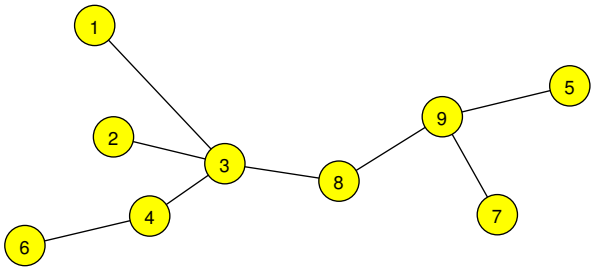
$\mathcal{A}_2$:



Determine for each state $p \in Q_1$ the set $Q_p := \{q \in Q_2;\ q \equiv p\}$. Decide whether $\mathcal{A}_1$ and $\mathcal{A}_2$ are equivalent.

Problem 4: Prüfer Sequence **4 Points**

(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.