Prof. Dr. Susanne Albers
Dr. Dimitrios Letsios
Lehrstuhl Theoretische Informatik
Fakultät für Informatik
Technische Universität München

Fall Semester
November 10, 2014

Randomized Algorithms

Exercise Sheet 5

Due: November 17, 2014

Exercise 1 (10 points)
Consider the problem of sorting a sequence $a_1, a_2, \ldots, a_n$ of n numbers. We say that A is a comparison-based sorting algorithm if it is based solely on making comparisons between the elements in order to sort the sequence. Use Yao's MinMax Principle in order to show that the expected running time of any Las Vegas such algorithm is $\Omega(n \log n)$.

Hint: Any deterministic comparison-based sorting algorithm can be modeled as a decision tree in which every node corresponds to a comparison made by the algorithm.

Complexity Class P/poly (Non-Uniform Polynomial Time)

A language $L \in \{0,1\}^*$ belongs to the complexity class **P/poly** if there exists a family of circuits $\{C_1, C_2, \ldots\}$ such that, for each n, C_n has n inputs and one output, the size of C_n is bounded above by a polynomial $p(n)$ of n and, for each input $x \in \{0,1\}^n$, C_n outputs 1 if $x \in L$ and 0, otherwise. It has to be noticed that the circuit C_n for n inputs may have very little resemblance to the circuit C_{n+1} for $n+1$ inputs.

In complexity theory, there is an alternative definition of the complexity class **P/poly** with algorithms that "take advice". An algorithm A is said to use advice a if for any input string of length n, it is also given an advice string $a(n)$. We say that A decides a language L with advice a if on an input x it uses the advice string $a(|x|)$ in order to decide whether x belongs to L. Uniform algorithms are those that use no advice strings at all, whereas non-uniform algorithms are those that use such advice.

Then, the complexity class **P/poly** can be also defined as follows. A language $L \in \{0,1\}^*$ belongs to **P/poly** if there exists a sequence $\{a_1, a_2, \ldots\}$ of advice strings, where the size of a_n is bounded above by a polynomial $p(n)$ of n, and an algorithm A with worst-case running time polynomial in n such that, for each $x \in \{0,1\}^n$, $A(x, a_n)$ accepts if $x \in L$ and it rejects, otherwise.

Exercise 2 (10 points)
Show that the two definitions that of the class **P/poly** are equivalent.

*Hint: Consider a language $L \in$ **NP** for which there exists a verification algorithm A. Cook and Levin showed that is is possible to build a circuit C of polynomial size which simulates algorithm A on an input x in a way that C is satisfiable iff there exists a certificate y such that $A(x, y)$ accepts.*

Exercise 3 (10 points)
Recall the complexity class **BPP** presented in class. It can be shown that if a language L belongs to **BPP**, then there exists a randomized algorithm A running in worst-case polynomial time which decides L with error probability strictly less than $(\frac{1}{2})^n$, for every $x \in \{0,1\}^n$. By using this fact, show that $\mathbf{BPP} \subseteq \mathbf{P/poly}$.

Exercise 4 (10 points)
We assign uniformly at random m persons to chairs of a round table with $n \geq m$ chairs. What is the expected number of persons without a direct neighbor on the left side and right side seats?