Prof. Dr. Susanne Albers
Dr. Suzanne van der Ster
Dario Frascaria
Lehrstuhl für Theoretische Informatik
Institut für Informatik
Technische Universität München

Fall Semester
November 16, 2015

Randomized Algorithms

Exercise Sheet 5

Due: November 23, 2015
at 10:15, in class

Exercise 5.1 (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 Minimax Principle in order to show that the expected running time of any Las Vegas comparison-based 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.

Exercise 5.2 (10 points)
Consider the following problem. Given a string $x \in \{0,1\}^n$, we want to determine if x contains two consecutive 1s.

By using Yao's Minimax Principle, show that the expected number of bits inspected by any randomized algorithm is $\Omega(n)$.

Exercise 5.3 (10 points)
Consider the Find-bill problem, that is stated as follows. There are n boxes and exactly one box contains a dollar bill. The other boxes are empty. A probe is defined as opening a box to see if it contains the dollar bill. The objective is to locate the box containing the dollar bill, while minimizing the number of probes performed. Consider the following randomized algorithm.

```
Select x ∈ {H, T} uniformly at random.
if x = H
    then  Probe boxes in order 1, ..., n and stop if bill is located
    else  Probe boxes in order n, ..., 1 and stop if bill is located
```

(a) Show that the expected number of probes for this algorithm equals $\frac{n+1}{2}$.

(b) Show that a lower bound on the expected number of probes required by any randomized algorithm to solve the Find-bill problem is $\frac{n+1}{2}$.

Exercise 5.4 (10 points)
In class, the concept of boolean circuits is introduced. Using only OR, AND and NOT gates, design a boolean circuit that correctly evaluates the following boolean expression

$$\neg\big((x_1 \oplus x_2 \oplus x_3) \land (x_4 \oplus x_5 \oplus x_6)\big).$$

Recall that $\oplus$ is the XOR (exclusive or) operator, only evaluating to 1 if exactly one of the two inputs is 1 and the other is 0.