Prof. Dr. Christian Scheideler
Universität Paderborn

Paderborn, April 20, 2018

Randomized Algorithms

SS 2018

Homework Assignment 2

Problem 5:
Consider an arbitrary decision problem P (i.e., there are only outputs of the form "YES" or "NO"). Suppose that we have a randomized algorithm A for P with the following property:

- For all inputs $x \in P$, $\Pr[A(x) = "NO"] \leq 1/3$ and
- for all inputs $x \notin P$, $\Pr[A(x) = "YES"] \leq 1/3$.

In other words, the error probability of A is at most $1/3$. Show that by executing A multiple times and using an appropriate decision rule based on its outputs, one can reduce the error probability to at most $1/n$.

Problem 6:
Consider the randomized Quicksort algorithm in Section 2.3. Let $X_{i,j}$ be defined as in Section 2.3 and let $X = \sum_{i<j} X_{i,j}$.

(a) Show that for any two distinct pairs $i < j$ and $i' < j'$ (i.e., i, j, i', j' are pairwise distinct) it holds that $\mathbb{E}[X_{i,j} \cdot X_{i',j'}] \leq \mathbb{E}[X_{i,j}] \cdot \mathbb{E}[X_{i',j'}]$ by going through all cases of (i, j) and (i', j').

(b) Use (a) (and the fact that for all non-distinct pairs $i < j$ and $i' < j'$, $\mathbb{E}[X_{i,j} \cdot X_{i',j'}] \leq min\{\mathbb{E}[X_{i,j}], \mathbb{E}[X_{i',j'}]\}$) to prove an upper bound on $\mathbb{V}[X]$, the variance of X.

(c) Use $\mathbb{V}[X]$ together with the Chebychev inequality (Theorem 1.8) to bound the probability of deviating from $\mathbb{E}[X]$. How large do we have to set k there to obtain a probability of at most $1/n$?

Problem 7:
Prove Lemma 2.9.

Problem 8:
In the well-known skip list data structure, a set of n elements is arranged in a set of sorted lists $L_0, L_1, L_2, \ldots$, where L_0 is the sorted list containing all elements. In addition to that, every element e_i chooses a random bit vector x_i, and element e_i participates in list L_k if and only if the first k bits in x_i are 1.

(a) What is the expected index of the highest list that element e_i participates in?

(b) Show that with high probability (i.e., a probability of at least $1 - 1/n$) the highest index k of a list L_k that contains elements is $O(\log n)$.

(c) Propose a search operation that would reach any element in $O(\log n)$ time when starting with the first element in L_0. (A formal analysis of the runtime is not needed.)