Prof. Dr. Benjamin Doerr **Winter 2012/13**

Exercises for Randomized Methods in Computer Science

http://www.mpi-inf.mpg.de/departments/d1/teaching/ws12/rmcs/

Assignment 1 Due: Tuesday, October 23, 2012

Exercise 1

Make yourself familiar with the notions of a discrete *probability space*, *independence of events*, *conditional probabilities*, *random variables*, *expectation*, and *independence of random variables*. Similarly, learn the following basic theorems: The *union bound*, the *law of total probability*, and *linearity of expectation*. All this can be found, e.g., on the first 23 pages of the Mitzenmacher/Upfal book.

Use the above material to give a formal proof of the results of the first lecture that a random assignment satisfies an expected number of 7/8 of the clauses of a Max-3-Sat instance.

Exercise 2 (*Coupon Collector*)

This is one of the basic tools, which often is very useful. Consequently, you can find the solution to this exercise in any textbook. But you learn more if you try yourself.

Imagine that there are n types of coupons. In each round, you get a coupon of a random type. How many rounds does it take until you have a coupon of each type? Try the following approach to obtain a pretty good upper bound.

Compute the probability p_r that after r rounds, you do not have a coupon of a fixed type. Then after r rounds the probability q_r that you do not have all coupons, is at most np_r. Show that this "failure probability" q_r is at most $n^{-\varepsilon}$ for $r = (1+\varepsilon)n\ln(n)$. In simple words, you can be quite sure that a little more than $n\ln(n)$ rounds suffice. For the last estimate, you might want to use an estimate so simple that one might be ashamed to use it (of course, no reason for that), namely that $1+x \leq e^x$ for all $x \in \mathbb{R}$.

Exercise 3 (*Mastermind with two colors*)

We have seen in the lecture that a good strategy in the Mastermind game with n holes and $k = n$ colors is to query random solutions. For two colors, this random guessing strategy is (i) the best that is known and (ii) optimal apart from possibly a constant factor of 2. As a first step towards proving this, prove the following.

Let $y,z \in \{0,1\}^n$ ("bit-strings", two-color codes). Denote by $d := d(y,z) := |\{i \in [n] \mid y_i \neq z_i\}|$ the Hamming distance of y and z, that is, the number of positions they differ in. Let x be a random bit-string (chosen uniformly from all length-n bit-strings). Compute the probability that x agrees with y and z in the same number of positions. In other words, this is the probability that a random query in the 2-color Mastermind game gives the same answer both for y and z being the secret code. No proof is asked for, it suffices if you give an expression (depending in d and n) that gives the correct probability.

Exercise 4

In this exercise, we will see a first (toy-)example showing that average-case complexity might not always tell us what we want. We consider the problem of computing a triangle in an undirected graph $G = (V,E)$. A triangle is a set of three vertices x, y, and z such that $\{x,y\},\{y,z\},\{z,x\} \in E$.

a) Argue that any deterministic algorithm computing a triangle takes time $\Omega(|E|)$.

A reasonable notion of average complexity of such an algorithm is the average run-time of the algorithm taken over all input instances, that is, over all graphs of a given size $n := |V|$. This is nothing else than the expected run-time on a random graph taking uniformly at random from the set of all graphs on the vertex set $V = \{1,\dots,n\}$. Note that such a random graph can be constructed by flipping a fair coin for each possible edge $\{x,y\}$ and including this edge into E when the coin shows the heads side. For this reason, let us consider the problem of finding a triangle in such a random graph.

b) Give an algorithm that surprisingly well finds a triangle in a random graph. Try to make this statement precise, and ideally, prove it. Note: This exercise is intentionally formulated not very precise, partially because we do not have all technical machinery, partially because different answers are possible.