Date: 2025-06-18
Version: 2025-06-17 18:30

Exercise Sheet 6 for Advanced Algorithms, Summer 2025

Hand In: *Until 2025-06-25 18:00, on ILIAS.*

Problem 1

20 + 30 points

We consider the two variants for Las Vegas algorithms: always-decisive LV and always-terminating LV.

a) Prove Theorem 7.7 from class:

Every Las Vegas algorithm A for $f : \Sigma^\star \to \Gamma^\star$ can be transformed into a always-decisive LV algorithm, i.e., a randomized algorithm B for f so that for all $x \in \Sigma^\star$ holds

(i) $\Pr[B(x) = f(x)] \;=\; 1$ (always correct)

(ii) $\mathbb{E}\text{-}\mathit{time}_B(x) \;\le\; 2 \cdot \mathit{time}_A(x)$

b) Prove Theorem 7.8 from class:

Every randomized algorithm B for $f : \Sigma^\star \to \Gamma^\star$ with $\Pr[B(x) = f(x)] = 1$ can be transformed into an always-terminating Las Vegas algorithm A for f so that for all $x \in \Sigma^\star$ holds

$$\mathit{time}_A(x) \;\le\; 2 \cdot \mathbb{E}\text{-}\mathit{time}_B(x).$$

Hint: Recall the *Markov's inequality*.

Problem 2

20 + 30 points

Let us consider the model of flipping a fair coin n times and denote by $X \in [0 : n]$ the total number of "heads" among the n coin flips.

a) For the concrete value $n = 100$, compute
   (i) the exact probability $\Pr[X \geq 66]$ (use computer algebra!),
   (ii) an upper bound for $\Pr[X \geq 66]$ using *Markov's inequality*,
   (iii) an upper bound for $\Pr[X \geq 66]$ using *Chebychev's inequality*, (recall the formula for $\mathrm{Var}[X]$), and
   (iv) an upper bound for $\Pr[X \geq 66]$ using the *Chernoff bound* for the binomial distribution.

b) Prove that we have for any $\varepsilon > 0$ that $X = \mathbb{E}[X] \pm \mathcal{O}(\sqrt{n}\log(n))$ w. h. p. as $n \to \infty$.

Problem 3

20 points

A one-sided-error Monte Carlo algorithm A for a function $f : \Sigma^\star \to \{0, 1\}$ might give a wrong answer every other time, but an answer $A(x) = 1$ is guaranteed to be correct. We could use majority voting to amplify the success probability $\frac{1}{2}$ to any desired constant, but that would not exploit the one-sidedness.

Describe how we can reduce the error probability to an arbitrary given constant $\delta > 0$, and compute the running time of the resulting method.

What is the running time to obtain a correct result *with high probability*? Compare your result to the majority-voting result from class for two-sided error Monte Carlo methods.