Komplexitätstheorie

Sommersemester 2019

Mario Holldack, M. Sc.
Prof. Dr. Georg Schnitger
Hannes Seiwert, M. Sc.

GOETHE
UNIVERSITÄT
FRANKFURT AM MAIN

Institut für Informatik
AG Theoretische Informatik

Übungsblatt 5

Ausgabe: 20.05.2019
Abgabe: 27.05.2019

Aufgabe 5.1 *Uniformes Ziehen mit fairen Münzen* (6 Punkte)

Entwerfen Sie einen Algorithmus A, der Zugriff auf eine Folge $b := (b_0, b_1, b_2, \dots)$ von fairen[1] Zufallsbits hat und die folgende Eigenschaft besitzt: Bei Eingabe einer Zahl $N \in \mathbb{N}_{>0}$ und einer Konstante δ mit $0 < \delta < 1$, beide in Binärdarstellung gegeben, gibt A in Worst-Case-Laufzeit $\texttt{poly}(\log N, \log \frac{1}{\delta})$ den Wert „FEHLER“ mit einer Wahrscheinlichkeit von höchstens δ aus. Falls A nicht „FEHLER“ ausgibt, gibt A eine uniform zufällig gezogene Zahl $i \in \{0, \dots, N-1\}$ aus, d. h. es gilt

$$\mathbf{pr}[A \text{ gibt i aus} \mid A \text{ gibt nicht „FEHLER“ aus}] = \tfrac{1}{N}.$$

Hinweis: Untersuchen Sie zunächst den Fall, dass N eine Zweierpotenz ist.

Aufgabe 5.2 *Faire Münzwürfe mit unfairen Münzen* (6 Punkte)

Sei $b := (b_0, b_1, b_2, \dots)$ eine Folge von unfairen, unabhängig und identisch verteilten Zufallsbits, d. h. es gilt $p := \mathbf{pr}[b_i = 1] \notin \{0, \frac{1}{2}, 1\}$ für alle $i \in \mathbb{N}$, wobei wir p nicht kennen. Entwerfen Sie einen Algorithmus A, der Zugriff auf die Folge b hat und ein faires Zufallsbit in erwarteter Laufzeit $\mathcal{O}(1/p(1-p))$ ausgibt.

Aufgabe 5.3 *Faktorisierung und Quadratwurzeln* $(4 + (2 + 6) = 12$ Punkte)

Für jedes $n \in \mathbb{N}_{>0}$ sei $\mathbb{Z}_n^*$ die Menge der *primen Restklassen* modulo n und QR_n die Menge der *quadratischen Reste* modulo n, wobei $\mathbb{Z}_n^* = \{i \in \{1, \dots, n-1\} : \mathrm{ggT}(i, n) = 1\}$ und $\mathrm{QR}_n = \{i^2 \bmod n : i \in \mathbb{Z}_n^*\}$. Eine *Quadratwurzel* von $c \in \mathrm{QR}_n$ ist eine Zahl $m \in \mathbb{Z}_n^*$ mit $m^2 \equiv c \bmod n$.

In dieser Aufgabe zeigen wir, dass die Bestimmung aller vier Quadratwurzeln eines quadratischen Restes $c \in \mathrm{QR}_n$ genauso schwierig ist wie die Faktorisierung von n.

a) Geben Sie einen effizienten Algorithmus an, der bei Eingabe einer Primzahl p mit $p \equiv 3 \bmod 4$ und eines quadratischen Restes $c \in \mathrm{QR}_p$ eine Quadratwurzel m von c ausgibt.

   Hinweis: Setzen Sie $m \equiv c^d \bmod p$ für einen geeigneten Exponenten $d \in \mathbb{N}$ und nutzen Sie den „Kleinen Fermat“: $a^{p-1} \equiv 1 \bmod p$ für alle $a \in \mathbb{Z}_p^*$.

b) i) Zum Aufwärmen: Geben Sie die vier Quadratwurzeln m_1, m_2, m_3 und m_4 von 4 in $\mathbb{Z}_{15}^*$ an.

   ii) Sei $n = pq$ das Produkt zweier verschiedener Primzahlen p und q. Sei A ein Algorithmus, der bei Eingabe von n und $c \in \mathrm{QR}_n$ die vier Quadratwurzeln m_1, m_2, m_3 und m_4 von c ausgibt. Geben Sie die Primfaktoren p und q in Abhängigkeit von m_1, m_2, m_3 oder m_4 an und begründen Sie die Korrektheit.

   Hinweis: Betrachten Sie die größten gemeinsamen Teiler $\mathrm{ggT}(m_i + m_j, n)$ und $\mathrm{ggT}(m_i - m_j, n)$ für geeignete $i, j \in \{1, 2, 3, 4\}$.

[1] Es gilt also $p := \mathbf{pr}[b_i = 1] = \frac{1}{2}$ für alle $i \in \{1, \dots, k\}$.