Randomisierte Algorithmen

Prof. Dr. Klaus Meer, Ameen Naif

Aufgabenblatt 3
Version 08.06.2016

Aufgabe 1.

Wir werfen 1000-mal eine ideale Münze. Wie groß ist die Wahrscheinlichkeit, dass mehr als 500-mal "Kopf" fällt? (Tipp: Benutzen Sie Chebyshev-Ungl.)

Aufgabe 2.

Wir haben zwei Computer R_I und R_{II} über einen Kommunikationskanal verbunden. Beide Computer haben einen großen Datenbestand, der jeweils als Bitfolge mit n Bit gespeichert ist. Wir wollen effizient testen, ob diese identisch sind. Zu diesem Zweck dürfen R_I und R_{II} Informationen mit einander austauschen. Als Kosten werden die Anzahl der übertragenen Bit gezählt.

Dazu werden die Bitfolgen als natürliche Zahlen $x, y \in \mathbb{N}$ interpretiert, R_I hat die Zahl x und R_{II} hat y. Gesucht ist ein Kommunikationsprotokoll mit dem beide Computer herausfinden können ob $x = y$ ist.

(a) Überlegen Sie, warum ein deterministischer Algorithmus alle n Bit übertragen müsste. (Bzw. mit Kompression $\Omega(n)$ Bit.)

Nun betrachten wir den randomisierten Algorithmus EQ_1: R_I wählt zufällig und gleichverteilt eine Primzahl p zwischen 1 und n^2. Dann sendet R_I die Zahlen p und $s = x \bmod p$ an R_{II}. Nun berechnet R_{II} seinen Rest $q = y \bmod p$ und vergleicht q und s. Bei $q = s$ antwortet R_{II} mit "gleich", ansonsten mit "ungleich".

(b) Wieviele Bit werden übertragen? Wieviele Bit sind zum Abgleich von einem Terabyte Daten nötig?

(c) Welcher Art ist der Algorithmus? (Las Vegas, Monte-Carlo Varianten, ...) Betrachten Sie dazu, in welchen Situationen eine falsche Antwort möglich wäre.

Wenn $x \neq y$ ist, kann durch eine schlecht gewählte Primzahl p trotzdem $x \bmod p = y \bmod p$ sein. Wir wollen nun die Fehlerwahrscheinlichkeit für diese Situation nach oben abschätzen. Der Ausgangspunkt ist:

$$\Pr(x \neq y \wedge x \bmod p = y \bmod p) = \frac{\text{Anzahl schlechter Primzahlen für } (x, y)}{\text{Anzahl Primzahlen } \leq n^2}. \quad (1)$$

(d) Sei $\mathrm{Prim}(m)$ die Anzahl der Primzahlen $\leq m$. Recherchieren Sie eine untere Schranke für $\mathrm{Prim}(n^2)$. (Hinweis: Primzahlsatz)

(e) Für die schlechten Primzahlen gilt $x \bmod p = y \bmod p$. Zeigen Sie, dass dafür $w = |x - y|$ ein Vielfaches von p sein muss (bzw. p ein Teiler von w).

(f) Warum ist $w \leq 2^n$?

(g) Schätzen Sie nun ab, wieviele Primzahlen als Teiler von w höchstens in Frage kommen. Betrachten Sie dazu die Primzahlzerlegung von $w = p_1^{i_1} \cdots p_k^{i_k}$ mit Primfaktoren $p_1, \ldots, p_k$ und zeigen Sie, dass mit $k = n$ (also n schlechte Primzahlen) schon $w \geq n! > 2^n$ wäre.

(h) Wie groß ist die Fehlerwahrscheinlichkeit höchstens (in Abhängigkeit von n)? Welche Fehlerwahrscheinlichkeit ergibt sich für einen Terabyte Daten?

(i) Wie kann die Fehlerwahrscheinlichkeit weiter abgesenkt werden?