1. Übungsblatt zu Grundzüge von Algorithmen und Datenstrukturen, WS 12/13

Prof. Dr. Markus Bläser, Radu Curticapean, Christian Engels
http://www-cc.cs.uni-sb.de/course/38/

Abgabe: Donnerstag, 25. Oktober 2012, 12:00 Uhr

Aufgabe 1.1 Gegeben sei ein sortiertes unendliches Array von reellen Zahlen

$$a_0 < a_1 < a_2 < \cdots < a_n < \cdots$$

sowie eine Zahl $x \in \mathbb{R}$. Es gilt $\lim_{n\to\infty} a_n = \infty$ und $a_0 = -\infty$. Wir definieren m als die Stelle im Array, für die $a_m \leq x < a_{m+1}$ gilt.

Zeigen Sie, wie man in $O(\log m)$ Schritten m finden kann. Die Variablen Ihres Algorithmus dürfen Werte der Größe $O(m)$ beinhalten und Sie dürfen annehmen, dass x in Zeit $O(1)$ mit Zahlen des Arrays verglichen werden kann.

Aufgabe 1.2 Gegeben sei ein sortiertes Array von n reellen Zahlen

$$a_1 < a_2 < a_3 < a_4 < \cdots < a_n$$

sowie ein weiteres sortiertes Array von $m \in o(n)$ reellen Zahlen

$$b_1 < \cdots < b_m.$$

Geben Sie einen Algorithmus an, der in $O(m \log(\frac{n}{m}))$ Schritten testet, ob die Mengen $A := \{a_1, \ldots, a_n\}$ und $B := \{b_1, \ldots, b_m\}$ disjunkt sind.

Hinweis: Es kann helfen, das „große“ Array in Blöcke der Größe $O(\frac{n}{m})$ zu unterteilen und m binäre Suchen in den Blöcken auszuführen.

In der Vorlesung wurde das Maschinenmodell so definiert, dass Variablen nur Zahlen *polynomieller* Größe speichern können. In den folgenden beiden Übungen wird untersucht, was „möglich“ ist, wenn man auf diese Einschränkung verzichtet.

Aufgabe 1.3 Zeigen Sie: Wenn in Variablen beliebig große natürliche Zahlen gespeichert werden können, so kann man auf Eingabe N und K den Wert $\binom{N}{K}$ in $O(\log N)$ Schritten berechnen. Die folgenden Hinweise könnten weiterhelfen:

- Überlegen Sie sich zunächst, wie man in $O(\log N)$ Schritten aus A und N den Wert A^N berechnen kann.
- Welche Form hat die Dezimaldarstellung von $(1000+1)^N$, wenn wir voraussetzen, dass $\binom{N}{K} < 1000$ für alle $K \in \mathbb{N}$ gilt?

Aufgabe 1.4 a) Zeigen Sie, dass man mit Variablen unbeschränkter Größe in $O(\log^2 N)$ Schritten aus N den Wert $N!$ berechnen kann. Nutzen Sie hierfür Aufgabe 1.3.

b) Zeigen Sie nun, dass man mit Variablen unbeschränkter Größe in $O(\log^3 N)$ Schritten einen echten Teiler von N finden kann (falls er existiert). Dies ist also ein effizientes Faktorisierungsverfahren.[1] — Sie dürfen ohne Beweis die Tatsache benutzen, dass der größte gemeinsame Teiler zweier natürlicher Zahlen $\leq N$ in $O(\log N)$ Schritten gefunden werden kann.

c) Was sagt dies über das RAM-Modell mit unbeschränkten Speicherzellen aus?

[1] Mit einer unbeschränkten RAM lassen sich also moderne kryptographische Systeme wie RSA brechen. Derartige Systeme fußen auf der Annahme, dass sich Zahlen nicht effizient in Primfaktoren zerlegen lassen. Hierbei bedeutet „effizient“ in Zeit $O(\log^c N)$, also in polynomieller Zeit in der Anzahl der Stellen der Eingabe.