UNIVERSITÄT DES SAARLANDES
Fachbereich 14 – Informatik

Prof. J. Buchmann
Stefan Neis

2. November 1995

3. Übung zu

"Algebra für Informatiker"

1. Entwickele einen "left-to-right" – Algorithmus zur schnellen Exponentiation, d.h. einen Algorithmus, der die Binärdarstellung des Exponenten von links nach rechts abarbeitet statt umgekehrt, wobei man folgende Formeln benutzt:

$$\begin{aligned} g^{2n} &= (g^n)^2 \\ g^{2n+1} &= (g^n)^2 \cdot g \end{aligned}$$

   Versuche dabei, Rekursion zu vermeiden, um so einen Algorithmus zu erhalten, der etwas schneller ist als der in der Vorlesung angegebene.

2. Nun wollen wir dies in die Praxis umsetzen: Auf den Rechnern des Cip-Pools ist in /home/stud/neis/LiDIA eine aktuelle Zwischenversion unserer "LiDIA-Bibliothek" installiert. Im File
   /home/stud/neis/LiDIA/src/interfaces/libI/bigint.c
   kann man sehen, wie unsere "power" – Funktion funktioniert. Implementiere analog eine Potenzierung für bigints in LiDIA, die Deinen Algorithmus aus Aufgabe 2 benutzt und vergleiche die Laufzeiten bei der Berechnung der 100-sten Potenz einer zufälligen 10-stelligen Zahl. Vergleiche außerdem mit der Laufzeit des naiven Algorithmus, der die Zahl hundertmal aufmultipliziert.
   Die Dokumentation zu LiDIA steht in
   /home/stud/neis/LiDIA/doc/dvi/lidiaman.dvi
   Hinweis: Wenn das Programm z.B. "test.cc" heißt, kann man es mit folgender Kommandozeile compilieren:
   ```
   g++ -O2 -I/home/stud/neis/LiDIA/include test.cc
   -L/home/stud/neis/LiDIA/lib/LiDIA/sparc7/c++ -lLiDIA -lm
   ```
   Für Zeitmessungen ist die LiDIA-Klasse "timer" sehr nützlich.

3. Betrachte die Halbgruppe $(\mathbb{Z}^{n\times n}, \cdot)$. Wir nennen zwei Matrizen $A, B \in \mathbb{Z}^{n\times n}$ äquivalent, wenn es eine Matrix $U \in \mathbb{Z}^{n\times n}$ gibt mit $\det C = 1$ und $A = UB$. Zeige, daß dies eine Äquivalenzrelation ist. Untersuche, ob die Äquivalenzrelation mit $\cdot$ verträglich ist.

Abgabe:

Donnerstag, 9. 11. 1995