Technische Universität München
Institut für Informatik
Prof. Dr. Ernst W. Mayr

WS 1998/99
Aufgabenblatt 8
11. Januar 1999

Praktikum Algorithmen-Entwurf

(Abgabetermin: Montag, den 18. Januar 1999, 24.^{00} Uhr)

Aufgabe 1 Randomisierter Primzahltest

Implementieren Sie den im Skript beschriebenen Algorithmus zur randomisierten Erkennung von Primzahlen. Ihr Programm sollte folgende Benutzerschnittstelle zur Verfügung stellen:

- Aufruf mit einem Kommandozeilenparameter, der die Anzahl der gewünschten Wiederholungen des Tests angibt.
- Die zu testende Primzahl wird von der Standardeingabe eingelesen (als zusammenhängender String von Ziffern). Führende Leerzeichen sollten hierbei ignoriert werden. Ferner wird angenommen, daß die Eingabe weniger als 1000 Stellen besitzt.
- Auf der Standardausgabe wird `yes` oder `no` ausgegeben, je nachdem, ob die eingelesene Zahl eine Primzahl ist oder nicht.

Hinweise

Die bei diesem Aufgabenblatt erstellten Algorithmen werden für das nächste Aufgabenblatt (Implementierung des RSA-Verfahrens) wieder benötigt. Aus diesem Grund empfiehlt es sich, die Programme modular aufzubauen (wie übrigens immer `;-)`) und beispielsweise die Teile

- erweiterter Euklidischer Algorithmus (Da auf diesem Blatt nur der „normale“ Euklidische Algorithmus benötigt wird, können Sie auch einfach auf die Methode gcd() der Klasse integer in LEDA zurückgreifen. Allerdings müßten Sie spätestens beim nächsten Blatt die erweiterte Version programmieren.),
- Exponentiation durch sukzessives Quadrieren und
- eigentlicher Primzahltest

als eigenständige Funktionen zu implementieren.

Als Eingabe können Sie die Dateien `p10.dat` bis `p500.dat`, sowie `np10.dat` bis `np500.dat` verwenden. Hierbei steht `p` für Primzahl (Die Bedeutung von `np` stellen wir als Übungsaufgabe). Die Zahl danach gibt an, wieviele Stellen die Binärdarstellung der Zahl ungefähr besitzt. Die Dateien `cm1.dat` und `cm2.dat` enthalten Carmichael-Zahlen. Ihr Programm rufen Sie dann beispielsweise so auf:

```
primetest 10 < p300
```