



Universität Freiburg  
Institut für Informatik  
PD Dr. A. Heinz  
PD Dr. S. Schuierer

Georges-Köhler-Allee, Geb. 051  
D-79110 Freiburg i. Br.  
Tel. (0761) 203-8164  
Tel. (0761) 203-8165  
Freiburg, 30. Oktober 2000

## Algorithmentheorie Übungsblatt 2

Abgabe bis Montag, 13. November 2000, 11:00 Uhr (in der Vorlesung)  
Besprechung am Mittwoch, 15. November, 2000, 11-13 Uhr

### **Aufgabe 1:** (2 + 2 Punkte) Randomisierter Primzahltest

- Benutzen Sie das logarithmische Exponentiationsverfahren, um nachzuweisen, daß die Identität  $7^{256} \equiv 1 \pmod{257}$  gilt.
- Ist die Zahl 127 prim oder zusammengesetzt? Verwenden Sie das randomisierte Primzahltestverfahren. Die Wahrscheinlichkeit, daß Sie die korrekte Antwort geben, soll größer gleich 90% sein.

### **Aufgabe 2:** (1 + 3 Punkte) Kryptographie

Ein *Zertifikat* bestätigt die Echtheit eines öffentlichen Schlüssels. Es enthält den Namen der ausgebenden Behörde, den Namen des Schlüsselinhabers und seinen öffentlichen Schlüssel. Es wird mit dem privaten Schlüssel der ausgebenden Behörde verschlüsselt oder signiert. Über den öffentlichen Schlüssel der Behörde kann es überprüft werden.

- Geben Sie ein Beispiel für einen Mißbrauch an, der durch Zertifikate verhindert werden kann.
- Alice möchte über das Internet mit ihrer Bank Kontakt aufnehmen. Sie kennt den öffentlichen Schlüssel der Bank noch nicht. Die Bank verfügt aber über ein Zertifikat einer Behörde, deren öffentlicher Schlüssel Alice bekannt ist. Geben Sie ein Protokoll an, mit dem die Bank Alice über eine Netzwerkverbindung ihre Identität beweisen kann. Versuchen Sie, möglichst viele Sicherheitsrisiken auszuschließen.

### **Aufgabe 3:** (2 + 2 Punkte) Skiplisten

Die Funktion *random()* liefere für die ersten 20 Aufrufe die folgende Sequenz von Werten aus  $\{0, 1\}$ : 00101100101101110001. Die Funktion *randomhöhe()* sei mit Hilfe von *random()* wie folgt definiert:

```

int randomhöhe ();
höhe = 0;
while random == 0 do
    höhe = höhe + 1;
return höhe

```

- a) Geben Sie die Liste an, die entsteht, wenn die Schlüssel 15, 2, 43, 17, 4, 8, 47, 28, 10, 92 der Reihe nach in die anfangs leere Skip-Liste eingefügt werden und zur Erzeugung der Höhen der Listenelemente die o.a. Funktion *randomhöhe()* verwendet wird.
- b) Berechnen Sie die gesamten Suchkosten, um auf jedes Element in der in Aufgabenteil a) erzeugten Skip-Liste genau einmal zuzugreifen. Vergleichen Sie diese Kosten mit den Suchkosten einer gewöhnlichen, linearen, aufsteigend sortierten Liste, die dieselben Schlüssel speichert.

**Aufgabe 4:** (4 Punkte) Randomisierte Suchbäume

Gegeben sei die (feste) Folge von Schlüssel  $F : 15, 2, 43, 17, 4, 8, 47$ .

Geben Sie den randomisierten Suchbaum für  $F$  an, der entsteht, wenn man die Schlüssel der Reihe nach in den anfangs leeren randomisierten Suchbaum einfügt und die Prioritäten der Schlüssel von einem Zufallszahlengenerator wie folgt erzeugt werden: 0.95, 0.25, 0.87, 0.32, 0.78, 0.01, 0.44. Was ist die Zahl der dabei insgesamt ausgeführten Rotationen?