2. Übungsblatt zu Grundzüge von Algorithmen und Datenstrukturen, WS 15/16

Prof. Markus Bläser
http://www-cc.cs.uni-saarland.de/course/50/

Abgabe: 5. November 2015, 12:00 Uhr

Aufgabe 2.1 In der Vorlesung Programmierung 1 haben sie Insertion-Sort kennengelernt. In dieser Aufgabe sollen sie nun Insertion-Sort in Pseudo-Code formulieren und analysieren.

a) Geben sie dazu eine Methode Insert(i, $a[1..n]$) an. Insert(i, $a[1..n]$) nimmt an, dass $a[n-i+1..n]$ bereits sortiert ist. Nach Aufruf von Insert(i, $a[1..n]$) soll $a[n-i..n]$ sortiert sein.

b) Formulieren sie nun Insertion-Sort unter Verwendung von Insert.

c) Formulieren sie eine Invariante für die Schleife von Insertion-Sort und beweisen sie damit die Korrektheit von Insertion-Sort. Sie dürfen annehmen, dass Insert korrekt arbeitet.

d) Schätzen sie die Laufzeit von Insertion-Sort ab.

Aufgabe 2.2 Die Lukas-Zahlen (L_n), $n \geq 0$, werden definiert als $L_0 = 2$, $L_1 = 1$ und $L_{n+1} = L_n + L_{n-1}$ für $n \geq 1$. Überlegen Sie sich, wie man mittels $O(\log n)$ arithmetischen Operationen die Lukas-Zahl L_n ausrechnen kann. Erlaubt sind in dieser Aufgabe nur Operationen mit *ganzen* Zahlen. Die Variablen Ihres Algorithmus dürfen Werte bis zu $2^{O(n)}$ beinhalten. (Insbesondere passt also die n-te Lukas-Zahl in eine einzelne Variable.)

Aufgabe 2.3 a) Entwerfen Sie einen Algorithmus, der gegeben zwei Arrays $a[1..n]$ und $b[1..n]$ entscheidet, ob das eine Array eine Permutation des anderen Arrays ist, d.h. ob es eine Permutation $\pi \in S_n$ gibt, so dass $a[i] = b[\pi(i)]$ für alle $1 \leq i \leq n$ gilt. Die Laufzeit Ihres Algorithmus sollte $o(n^2)$ sein.

b) Erweitern Sie den Algorithmus so, dass er zusätzlich eine solche Permutation π findet.