Einführung Informatik / Algorithmen und Datenstrukturen WS 97/98
Prof. Dr. R. Kruse / I. Blümel / C. Borgelt

3. Übungsblatt
Thema: Effizienz/Komplexität von Algorithmen

Aufgabe 18

Mit Hilfe des *Siebs des Eratosthenes* können sehr effizient alle Primzahlen bis zu einer gegebenen Obergrenze gefunden („herausgesiebt“) werden. Stellen Sie sich eine Tafel vor, auf der alle natürlichen Zahlen bis zu einer gewählten Obergrenze angeschrieben sind. Auf dieser werden folgendermaßen alle Nicht-Primzahlen gestrichen: Beginnen Sie mit der 2. Alle Vielfachen von 2, beginnend mit 4, sind keine Primzahlen. Streichen Sie diese Zahlen auf der Tafel aus. Nehmen Sie nun die 3. Streichen Sie alle Vielfachen von 3, beginnend mit 6. Nun die 4. Sie ist bereits von der Tafel gestrichen und kann daher übersprungen werden. (Alle Vielfachen von 4 sind auch Vielfache von 2 und daher schon gestrichen.) Streichen Sie die Vielfachen von 5, beginnend mit 10. Indem Sie so fortfahren und am Ende noch die 1 streichen, die per Definition keine Primzahl ist, bleiben auf der Tafel nur die Primzahlen ungestrichen. Schreiben Sie ein Programm, das diesen Algorithmus benutzt, um alle Primzahlen bis 1000 zu bestimmen.

Aufgabe 19

Berechnen Sie für die Registermaschine M aus Beispiel 3.8 (Primzahltest) die Zeit- und Raumkomplexitäten für die Eingaben 77 und 37, die Zeit- und Raumkomplexitäten des schlechtesten Falles $T_M(n)$ und $S_M(n)$ und die durchschnittliche Zeitkomplexität für Eingaben der Größe 2 (bei Verwendung der Dezimaldarstellung).

Aufgabe 20

Gegeben seien die folgenden Kursänderungen einer Aktie über 10 Wochen: +3.1, −4.1, +5.9, +2.6, −5.3, +5.8, +9.7, −9.3, −2.3, +8.4. Bestimmen Sie den maximalen Gewinn, den Sie durch einen einmaligen Kauf und Wiederverkauf einer Aktie dieses Typs innerhalb des betrachteten 10-Wochen-Zeitraums hätten erzielen können.

In mathematischer Formulierung ist dies das Problem der maximalen Teilsummen. Gegeben sei eine Folge $F = (a_1, a_2, \dots, a_n)$ reeller Zahlen. Nehmen Sie zur Vereinfachung an, daß mindestens eine dieser Zahlen positiv ist. Gesucht ist die maximale Teilsumme dieser Folge, d.h. das Maximum der Summen von Teilfolgen der Folge F. Eine Teilfolge der Folge F ist eine Folge $F_{ij} = (a_i, a_{i+1}, \dots, a_j)$ mit $1 \leq i \leq j \leq n$. Wir suchen also

$$\mathrm{mts}(F) = \max_{1 \leq i \leq j \leq n} \sum_{k=i}^{j} a_k.$$

Implementieren Sie die verschiedenen in der Vorlesung behandelten Algorithmen zur Lösung dieses Problems. Lassen Sie Ihre Programme nach jedem Berechnungsschritt ein Zeichen (z.B. einen Stern `*`) ausgeben. Sie können dann an der unterschiedlichen Zahl ausgegebener Zeichen die unterschiedliche Zeitkomplexität ablesen.

Zusatzaufgabe (für besonders Eifrige): Lassen Sie sich von Ihrem Tutor zeigen, wie man Zahlen aus einer Datei einliest bzw. wie man eine Datei als Eingabe eines Programms verwenden kann. Testen Sie dann Ihre Programme auf den Dateien `n1000.dat` und `n10000.dat`, die Sie im HP-Pool im Verzeichnis `/users/iik/borgelt/data` finden.