



7. ÜBUNG ZUR VORLESUNG PROGRAMMIERUNG II, SS 2003

Aufgabe 1: Rekurrenz, Substitutionsmethode (10 Punkte)

Lösen Sie die folgenden Rekurrenzgleichungen mit der *Substitutionsmethode*:

- $T(n) = T(\lfloor \log_2(\sqrt{n}) \rfloor) + 12$
- $T(n) = 4T(\lfloor \frac{n}{2} \rfloor) + n$
- $T(n) = 3T(\lfloor \frac{n}{4} \rfloor) + 2T(\lceil \frac{n}{8} \rceil) + 4$

Aufgabe 2: Rekurrenz, Mastertheorem (10 Punkte)

Welche der folgenden Rekurrenzgleichungen können Sie mit dem *Master-Theorem* auswerten? Begründen Sie Ihre Antwort, und bestimmen Sie in den Fällen, in denen das Theorem anwendbar ist, die Lösung der Gleichung.

- $T(n) = 36T(\frac{n}{6}) + \frac{n^2}{\ln(n)}$ (für $n > 1$)
- $T(n) = 2\sqrt{2}T(\frac{n}{2}) + 42\sqrt{n^3} + 4711$
- $T(n) = 125T(\frac{n}{5}) + n^2\sqrt{n}$
- $T(n) = 1000000T(\frac{n}{10}) + \frac{1}{4711}e^n$
- $T(n) = 27T(\frac{n}{3}) + 12n^3\log^2(n)$

Aufgabe 3: Markov- und Chebychev Ungleichung (10 Punkte)

Eine Messung der Schuhgrößen von 55048 englischen Schülern zwischen 7 und 16 Jahren hat folgende Daten ergeben (<http://censusatschool.ntu.ac.uk/results.asp>):

Größe	15 cm	18 cm	20 cm	22 cm	24 cm	26 cm	28 cm
Anzahl	1509	5139	12747	15838	12441	5336	2038

Berechnen Sie den Erwartungswert der Schuhgröße eines englischen Schülers und bestimmen Sie mit Hilfe der Markov Ungleichung eine obere Schranke für die Wahrscheinlichkeit, dass die Schuhgröße eines Schülers um mehr als 20% von diesem Wert abweicht. Verbessern Sie dann Ihre Abschätzung mit Hilfe der Chebychev Ungleichung.

Aufgabe 4: Fibonacci-Zahlen (10 Punkte)

- (a) Beschreiben Sie einen rekursiven Algorithmus, der die n -te Potenz a^n in $\Theta(\log_2(n))$ berechnet. Beweisen Sie diese Laufzeitangabe für Ihren Algorithmus.
- (b) Die *Fibonacci-Zahlen* $\text{Fib}(n)$ sind definiert durch

$$\text{Fib}(0) = 0, \text{Fib}(1) = 1, \text{Fib}(n+2) = \text{Fib}(n+1) + \text{Fib}(n), n \geq 0$$

Fasst man dies als (naiven) rekursiven Algorithmus auf, so ergibt sich für die Berechnung von $\text{Fib}(n+2)$ eine Laufzeit von $\Theta(2^n)$.

- (i) Zeigen Sie:

$$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} \text{Fib}(n-1) & \text{Fib}(n) \\ \text{Fib}(n) & \text{Fib}(n+1) \end{pmatrix} = \begin{pmatrix} \text{Fib}(n) & \text{Fib}(n+1) \\ \text{Fib}(n+1) & \text{Fib}(n+2) \end{pmatrix}$$

- (ii) Beschreiben Sie einen rekursiven Algorithmus, der $\text{Fib}(n+2)$ in $\Theta(\log_2(n))$ berechnet.

Hinweis: Überlegen Sie sich, wie Sie die linke Seite von Teil (b) weiter rekursiv zerlegen können, bis Sie es nur noch mit “trivialen” Faktoren zu tun haben.

Zusatzaufgabe 5: Fibonacci-Zahlen, Implementierung (10 Bonuspunkte)

Implementieren Sie eine Templateklasse `TMatrix<T>` für beliebige $n \times m$ - Matrizen mit Einträgen vom Typ T . Ihre Klasse sollte ein möglichst vollständiges Interface inklusive Default- und Copy-Konstruktor und Destruktor sowie einem Zuweisungsoperator enthalten. Implementieren Sie die Addition, Subtraktion und Multiplikation (durch Implementieren der Operatoren `+`, `-`, `*`) sowie die Exponentiation und die Multiplikation mit einem Skalar (also einer Variable vom Typ T).

Schreiben Sie ein Programm, das mit Hilfe dieser Klasse die n -te Fibonacci-Zahl bestimmt, wobei n als Kommandozeilenparameter übergeben werden soll.

Kommentieren Sie Ihren Code ausführlich und *testen* Sie alle implementierten Funktionen (insbesondere auf das Verhalten in Grenzfällen). Kommentieren Sie auch Ihre Testfälle.

Die Abgabe erfolgt sowohl elektronisch (d.h. per email an Ihren Übungsgruppenleiter), als auch in Papierform. Fügen Sie auch ein Makefile bei, so dass durch Eingabe von `make aufgabe5` Ihr Quellcode zu einem Programm namens `fibonacci` kompiliert wird.

Abgabe: 13.06.2003