Grundzüge von Algorithmen und Datenstrukturen, WS 15/16: Lösungshinweise zum 1. Übungsblatt

Prof. Markus Bläser

Aufgabe 1.1

- Sei $f \in \mathrm{O}(g)$. Nach Definition gibt es dann ein $c > 0$ und ein n_0, so dass für alle $n > n_0$ gilt: $f(n)/g(n) < c$. Also ist für $n > n_0$ der Wert c eine obere Schranke der Menge $\{f(k)/g(k),\ k \geq n\}$, also ist $\lim_{n\to\infty} \sup_{k\geq n} f(k)/g(k) \leq c$.

- Gelte $\limsup_{n\to\infty} f(n)/g(n) < c$ für ein $c > 0$. Das heißt, dass

$$\lim_{n\to\infty} \sup_{k\geq n} f(k)/g(k) = c'$$

für ein $c' < c$. Daher gibt es für $\varepsilon = 1$ ein n_1, so dass für alle $n \geq n_1$ gilt:

$$|\sup_{k\geq n}\{f(k)/g(k)\} - c'| < 1.$$

Also ist für alle $n > n_1$ der Wert $c' + 1$ eine obere Schranke für die Menge $\{f(k)/g(k), k \geq n\}$. Also gilt für alle $k > n_1$

$$f(k) < (c+1)g(k).$$

- Da f und g in die *positiven* reellen Zahlen abbilden, ist die Definition der o-Notation äquivalent dazu, dass 0 Grenzwert von $f(n)/g(n)$ für $n \to \infty$ ist: Für alle $c > 0$ gibt es ein n_0, so dass für alle $n \geq n_0$ gilt $f(n) \leq cg(n)$ bzw., äquivalent dazu, $f(n)/g(n) \leq c$.

Aufgabe 1.2 (Lösungshinweise) Die Reihenfolge von langsam nach schnell ist:

$$7n \log n, n^2, n^{\log n}, 3^n, n!$$

Man kann z.B. Aufgabe 1 verwenden und die entsprechenden Grenzwerte ausrechnen. $n!$ lässt sich mit der Stirlingschen Formel abschätzen. Hier reicht aber schon die grobe Abschätzung $(\frac{n}{2})^{\frac{n}{2}} \leq n! \leq n^n$.

Aufgabe 1.3 (Lösungshinweise) Wir benutzen „schnelles Potenzieren“. Die Idee ist, A zu quadrieren und N zu halbieren, falls N gerade ist. Falls N ungerade ist, wird A einmal zum Zwischenergebnis hinzumultipliziert. Da so N mindestens jeden zweiten Schritt gerade ist, halbiert sich N mindestens jeden zweiten Schritt, erreicht also nach spätestens $2\lceil \log N \rceil$ Schritten den Wert 1.

Wegen

$$(1000+1)^N = \sum_{0\leq i\leq N} \binom{N}{i} 1000^i$$

sind jeweils drei Dezimalstetllen von 1001^N gerade die Binomialkoeffizienten $\binom{N}{0}$, $\binom{N}{1}$, $\ldots$, $\binom{N}{N}$ – wenn diese Binomialkoeffizienten nicht mehr als drei Stellen haben. So kann man mit DIV und MOD beliebige Binomialkoeffizienten $\binom{N}{K}$ aus $(A+1)^N$ berechnen, wenn man A groß genug wählt.

Aufgabe 1.4 (Lösungshinweise) Da Binomialkoeffizienten etwas mit Fakultät zu tun haben, versuchen wir die Fakultät mit Hilfe von Binomialkoeffizienten zu berechnen. Das geht z.B. wie folgt. Wegen

$$\binom{2k}{k} = \frac{(2k)!}{(k!)^2}$$

haben wir für gerade n die Rekursionsgleichung $n! = \binom{n}{n/2} \cdot \left(\frac{n}{2}\right)!$. Für ungerade n können wir $n! = n \cdot (n-1)!$ benutzen.

Es gilt $\gcd(1!, N) = 1$ und $\gcd(N!, N) = N$. Wenn N einen nichttrivialen Teiler hat, dann insbesondere einen kleinsten Primteiler P, $2 \leq P < N$, für den dann $1 < \gcd(P!, N) = P < N$ gilt. Wir haben immer $\gcd(K!, M) \leq \gcd(L!, M)$ für $K \leq L$. Daher können wir mittels binärer Suche auf $\{1, 2, \ldots, N\}$ ein T finden, für das $1 < \gcd(T!, N) < N$ gilt. Jedes solche T ist ein nichttrivialer Teiler von N.

Für die Berechnung von Ausdrücken der Form $\gcd(T!, N)$ überlegen wir uns, dass $\gcd(T!, N) = \gcd(T! \bmod N, N)$ ist. Die Parameter auf der rechten Seite sind beide höchstens N, also kann die rechte Seite mit dem euklidischen Algorithmus in $O(\log N)$ Schritten bestimmt werden. Den Wert $T! \bmod N$ können wir mittels einer Division mit Rest aus $T!$ bestimmen bzw. der euklidische Algorithmus zur Bestimmung von $\gcd(T!, N)$ erledigt dies automatisch als ersten Schritt.