Algorithmen und Datenstrukturen
Übungsblatt 1

Prof. Dr. U. Schöning
T. Büchler
Institut für Theoretische Informatik

Aufgabe 1.1 *3 Punkte.*

Für eine Aufgabenstellung wurden verschiedene Algorithmen mit folgendem Berechnungsaufwand ermittelt (angegeben ist jeweils die Anzahl der Operationen für die Eingabelänge $n \in \mathbb{N}$).

$$(a)\ 50000000n, \quad (b)\ 10^7 n \log_2 n, \quad (c)\ 10^6 n^2, \quad (d)\ 1{,}5^n, \quad (e)\ 2^n, \quad (f)\ n!$$

Die Algorithmen sollen auf einem Rechner ausgeführt werden, der 10^9 Operationen pro Sekunde ausführen kann. Bis zu welcher Eingabelänge können die Algorithmen jeweils eingesetzt werden, wenn nach einer Sekunde eine Antwort vorliegen muss? Was ergibt sich, wenn man eine Minute Zeit hat?

Aufgabe 1.2 *6 Punkte.*

Zeigen oder widerlegen Sie die folgenden Aussagen ($n \in \mathbb{N}$ und $f, g : \mathbb{N} \to \mathbb{N}$).

(a) $\log(n^n) = \mathcal{O}(\, n \log n\,)$

(b) $\log_a n = \Theta(\, \log_b n\,)$ für reellwertige $a > 1$, $b > 1$.

(c) $2^{\log(n^2)} = \Omega(\, \sqrt{n^3}\,)$

(d) $\log(f(n)) = \mathcal{O}(\, \log(g(n))\,)$ für $f(n) = \mathcal{O}(\, g(n)\,)$

(e) $2^n = \mathcal{O}(\, (\sqrt{2})^n\,)$

(f) $f(n) = \log(n)^{\log(n)}$ ist polynomiell beschränkt.
(Hinweis: Eine Funktion $f : \mathbb{N} \mapsto \mathbb{R}_+$ heißt polynomiell beschränkt gdw. $\exists k \in \mathbb{N} : f(n) = \mathcal{O}(\, n^k\,)$.)

Aufgabe 1.3 *3 Punkte.*

Geben Sie die entsprechenden Laufzeiten in Θ-Notation für die gegebenen Rekursionsgleichungen an. Verwenden Sie dazu das Mastertheorem.

(a) $T_1(n) = 5 \cdot T_1(\frac{n}{3}) + T_1(\frac{2n}{3}) + 3 \cdot n$

(b) $T_2(n) = T_2(\frac{n}{4}) + 2 \cdot T_2(\frac{n}{16}) + \sqrt{n}$

(c) $T_3(n) = T_3(\frac{3n}{4}) + 2 \cdot T_3(\frac{n}{16}) + 4 \cdot n$

Aufgabe 1.4 *3 Punkte.*

Gegeben sei die Rekursionsgleichung

$$f(n) = 2 \cdot f(n-1) + f(n-2)$$

mit $f(1) = 1$ und $f(2) = 2$. Finden Sie eine obere Schranke der Form $f(n) \leq \alpha^n$. Geben Sie dabei ein möglichst kleines α an.

Aufgabe 1.5 *2+3 Punkte.*

Bei dem bekannten Spiel *Türme von Hanoi* soll ein Stapel unterschiedlich großer Holzscheiben von einem Startstapel zu einem Zielstapel bewegt werden. Dabei sind folgende Einschränkungen zu beachten:

- In der Ausgangsstellung liegen alle Scheiben mit absteigender Größe von unten nach oben sortiert auf dem Startstapel.
- Im Laufe des Spiels darf nie eine größere auf einer kleineren Scheibe zu liegen kommen.
- Es darf pro Zug nur eine Scheibe bewegt werden.
- Es darf von jedem Stapel immer nur die oberste Scheibe entfernt und auf einen anderen Stapel gelegt werden.
- Zusätzlich zum Start- und Zielstapel gibt es nur noch einen weiteren Ablagestapel.

a) Stellen Sie eine Rekursionsgleichung für die Zahl der minimal nötigen Bewegungen zum Lösen des Spiels auf. Bestimmen Sie für n gegebene Scheiben explizit die Anzahl der minimal benötigten Schritte zum Lösen des Spiels.

b) Lösen Sie die Programmier-Aufgabe Hanoi. Es ist zu langsam, das Verschieben der einzelnen Scheiben zu simulieren! Benutzen Sie daher das Ergebnis aus Teilaufgabe a), um die Lösung effizienter zu bestimmen.

Geben Sie Ihren Domjudge-Teamnamen bei Ihrer Abgabe an, damit Ihnen Ihre Lösung zugeordnet werden kann.