Dr. S. Lewandowski **Einführung in die Informatik I (autip)** WS 2007/08

Übungsblatt 07

Ausgabe: 05.12. Abgabeschluss: Mittw., 12.12., 9:45 Uhr, `eClaus.informatik.uni-stuttgart.de`

Abgabe erfolgt ausschließlich elektronisch über `eClaus.informatik.uni-stuttgart.de` – versuchen Sie nach Möglichkeit die Abgabe nicht in der letzten Minute zu machen!

Von jedem Aufgabenblatt werden maximal 20 Punkte auf den Schein angerechnet.

---

1. (3 Punkte, mittel) **Wald von Paganovo:** Im Zauberwald von Paganovo gibt es einen höchsten Baum mit Höhe $h_{\max}$. Alle weiteren Bäume gehorchen dem folgenden magischen Gesetz: Einen Baum der Höhe h gibt es im Wald nur dann, wenn es auch einen mit der Höhe $2(h + 13)$ oder einen mit Höhe $h + 37$ gibt. Natürlich hat jeder Baum eine positive Höhe. Schreiben Sie eine Ada-Funktion `Kleinster(h: Float) return Float`, die aus der Höhe des größten Baumes die des kleinsten Baumes berechnet. Bauen Sie diese Funktion in ein Hauptprogramm ein, das die Höhe des größten Baumes einliest und zum Schluss das Ergebnis ausgibt.

2. (12(+6) Punkte, mittel–schwer) **Kleingeld:** In der EU haben wir uns inzwischen an die üblichen Cent-Münzen zu 1, 2, 5, 10, 20 und 50 Cent gewöhnt. In anderen Ländern gibt es zum Teil andere Abstufungen (z.B. 1, 5, 10, 25 und 50 Cent in USA). Die gewöhnlichen Abstufungen haben alle die Eigenschaft, dass man die minimal benötigte Münzanzahl zur Darstellung eines vorgegebenen Centbetrags leicht ermitteln kann, indem man jeweils die größtmögliche Münze möglichst oft nimmt und den Rest auf die gleiche Art und Weise mit den kleineren Münzen auffüllt („gierige Methode“). Beispiel: 94 Cent – 50 Cent einmal (Rest 44), 20 Cent zweimal (Rest 4), 10 Cent nicht (Rest 4), 5 Cent nicht (Rest 4), 2 Cent zweimal (Rest 0), 1 Cent nicht. Lässt man beliebige Abstufungen zu, gilt dies nicht mehr. Z.B. bei der Aufteilung 1, 7, 11, 47, 58 und 70 würde man so 25 Cent in zweimal 11 Cent und dreimal 1 Cent aufteilen (gesamt 5 Münzen), wohingegen 3 Münzen (einmal 11 Cent und zweimal 7 Cent) ausreichen. Um hier die minimale Anzahl der zur Darstellung benötigten Münzen zu bestimmen, muss man also möglichst systematisch alle sinnvollen Kombinationen ausprobieren.

   - (1 Punkt, leicht) Schreiben Sie eine Funktion, die vom Benutzer die 6 verschiedenen zu benutzenden Münzwerte einliest. Achten Sie darauf, dass die Eingabe in aufsteigender Reihenfolge geschieht oder sortieren Sie nach Eingabe die 6 Zahlen. Stellen Sie sicher, dass eine 1-Cent-Münze vorgesehen wird.
   - (1 Punkt, leicht) Schreiben Sie eine Ada-95-Funktion, die zu einem gegebenen Centbetrag über die gierige Methode die dann nötige Anzahl der Münzen bestimmt. Berechnen Sie mithilfe dieser Funktion, wieviel Münzen zur Darstellung von 1 bis 99 Cent im Mittel benötigt werden.
   - (3 Punkte, mittel) Um alle sinnvollen Kombinationen zu testen, kann man für alle möglichen Anzahlen einer Münze den jeweiligen Restbetrag mit nur noch kleineren Münzen versuchen darzustellen. Im obigen Beispiel würde man, um 25 Cent aufzuteilen, die 11-Cent-Münze entweder zweimal verwenden (und die verbleibenden 3 Cent mit 7- und 1-Cent-Münzen darstellen) oder einmal (und 14 Cent anders aufteilen – und hier im Gegensatz zur gierigen Methode dann auch den minimalen Wert finden) oder garnicht (und die 25 Cent nur mit den kleineren Münzen darstellen). Schreiben Sie eine Ada-95-Funktion, die diese Strategie benutzt, und berechnen Sie auch hier mit deren Hilfe die im Mittel nötige Anzahl der Münzen zur Darstellung von 1 bis 99 Cent.

     Wenn Sie eine Kombination von 6 Münzwerten finden, die im Mittel weniger Münzen zur Darstellung von 1 bis 99 Cent benötigt als eine der beiden üblichen (1, 2, 5, 10, 20, 50 oder 1, 2, 5, 10, 25, 50), geben Sie diese mit dem gefundenen Mittelwert als Kommentarzeile mit an.
   - (4 Punkte, mittel–schwer) Ein weiterer Ansatz, der das korrekte Ergebnis findet, geht folgendermaßen vor: Wir speichern in einem Array A (Indizes von 1..99) jeweils ab, mit wieviel Münzen wir den Betrag bisher minimal darstellen konnten. Initialisiert wird dies mit A(i):=i für alle i (Darstellung von i mit i 1-Cent-Münzen) außer für die eingegebenen Münzwerte m_x, für die $A(m_{1,\ldots,6}):=1$ gesetzt wird. Nun wird das Feld immer wieder durchlaufen: im zweiten Durchlauf werden für die Einträge i, für die A(i)=1 ist, $A(i+m_x):=2$ gesetzt (außer es gilt $A(i+m_x)<2$), im dritten Durchlauf werden für die Einträge i, für die A(i)=2 ist, $A(i+m_x):=3$ gesetzt (außer es gilt $A(i+m_x)<3$), usw. Überlegen Sie, wie oft Sie auf diese Weise das Array durchlaufen müssen.

Zusatzaufgabe (für Freaks): Hier wird nur die nötige Anzahl bestimmt. Überlegen Sie, wie Sie die Funktion (ohne übermäßige Verlängerung der Laufzeit) erweitern müssen, um auch die zugehörige Zusammensetzung ausgeben zu können.

- (3 Punkte, schwer) Geben Sie als Kommentarzeilen mit Begründung an, welche der Varianten am schnellsten und welche am langsamsten ist.
- **Zusatzaufgabe (2 Punkte, mittel):** Es sollen nun die Anzahl der Münzen minimiert werden, die beim Bezahlen eines Centbetrags (1 bis 99) minimal ausgetauscht werden müssen (ohne Berücksichtigung einer evtl. vorhandenen 100-Cent-Münze). Bei 39 Cent werden dies z.B. 3 Münzen sein (zwei 20-Cent-Münzen hin und eine 1-Cent-Münze zurück). Testen Sie Ihr Programm u.A. mit 99 Cent.
- **Zusatzaufgabe (4 Punkte, schwer):** Lassen Sie durch ein Programm ermitteln, bei welchen 6 Münzwerten der Mittelwert der benötigten Münzen mit der gierigen Methode minimal wird und bei welchen 6 der Mittelwert bzgl. einer der exakten Verfahren minimal wird.

  Warnhinweis: Die Laufzeit für solch ein Programm kann sehr schnell sehr groß werden. Lassen Sie zum Programmtesten daher das Programm zunächst für nur 3 Münzwerte laufen (wobei 1 Münzwert fest auf der 1 bleibt) und betrachten Sie nur die Centbeträge 1 bis 25.

3. (8(+2) Punkte, mittel–schwer) **Verflixte Puzzelei:** Beim Aufräumen haben Sie im Keller ein Puzzle gefunden. Die 9 Teile sollen so zu einem Quadrat zusammengefügt werden, dass an jeder Kante ein durchgehender Pfeil und ggf. eine komplette geometrische Figur entsteht. Nach etwas Probieren haben Sie immerhin 8 Teile passend zusammengefügt, nur das Teil mit der Nummer 7 passt nicht in die verbleibende Ecke.
   (8 Punkte) Schreiben Sie ein Ada 95 Programm, das alle möglichen Lösungen des Puzzles ausgibt (geben Sie für die 3 Reihen jeweils an, welche Teile dort verwendet und ob diese um 90, 180 oder 270 Grad gedreht wurden).
   Zusatzaufgabe (2 Punkte): Erweitern Sie Ihr Programm, so dass Lösungen, die durch Drehung aus anderen Lösungen hervorgehen, nicht ausgegeben werden.