

38. Stabile Sortiervverfahren, 10 Punkte

Ein Sortieralgorithmus ist *stabil*, wenn es gleiche Werte in derselben relativen Reihenfolge belässt, in der sie in der Eingabe stehen. Bewerten Sie die Verfahren (i) Bubble-sort (ii) Sortieren durch Verschmelzen (iii) Sortieren durch Einfügen und (iv) Quick-sort bezüglich ihrer Stabilität. Es stehen folgende Antworten zur Auswahl.

- (a) Man kann gar nicht vermeiden, dass das Verfahren stabil ist.
- (b) Das Verfahren ist stabil, wenn man bei der Programmierung darauf achtet.
- (c) Das Verfahren kann nur mit Mühe stabil gemacht werden.

Begründen Sie Ihre Antworten. Im Fall (b) geben Sie Beispiele, was man falsch machen kann. Im Fall (c) geben Sie ein Beispiel, wo der Algorithmus nicht stabil ist.

39. Umrechnungstabelle, Programmieraufgabe, 10 Punkte

(8 Punkte) Schreiben Sie ein Programm, das bei Eingabe einer Währung und eines Wechselkurses eine schlaue Umrechnungstabelle für eine Auslandsreise nach folgendem Muster berechnet, und das *ohne Sortieren* auskommt. Alle Beträge, die zwischen 1 und 100 Euro ausmachen und die in einer der Währungen eine "runde Zahl" ergeben, sollen darin enthalten sein. (Die Aufteilung in drei Spalten ist nicht notwendig.)

313.81 HUF = 1.00 EUR	2000.00 HUF = 6.37 EUR	10000.00 HUF = 31.87 EUR
500.00 HUF = 1.59 EUR	3138.12 HUF = 10.00 EUR	15690.60 HUF = 50.00 EUR
627.62 HUF = 2.00 EUR	5000.00 HUF = 15.93 EUR	20000.00 HUF = 63.73 EUR
1000.00 HUF = 3.19 EUR	6276.24 HUF = 20.00 EUR	31381.20 HUF = 100.00 EUR
1569.06 HUF = 5.00 EUR		

(2 Punkte) Wieviele Einträge enthält die Tabelle höchstens und mindestens?

(Zusatzfrage, 0 Punkte.) Bei welchem Wechselkurs wird die größte relative Lücke (das heißt, das Verhältnis zwischen aufeinanderfolgenden Einträgen) am kleinsten?

40. Logarithmen, 0 Punkte. In der Vorlesung wurde gezeigt, dass Quicksort im durchschnittlichen Fall weniger als $2(n+1) \cdot H_n \leq 2(n+1)(1 + \ln n) = O(n \log n)$ Vergleiche benötigt. Warum muss man beim Logarithmus in der O -Notation nicht angeben, zu welcher *Basis* der Logarithmus gemeint ist?

41. Sortieren durch Verschmelzen „von klein auf“ (*bottom-up mergesort*), 10 Punkte

Bei diesem Verfahren wird nicht rekursiv zerlegt, sondern es werden immer längere sortierte Teillisten aufgebaut. Nach i Schritten hat man Teillisten der Länge 2^i , bis auf eine letzte Liste, die möglicherweise kürzer ist. Diese Listen werden dann paarweise zusammengefasst und verschmolzen.

- (a) Untersuchen Sie die exakte Anzahl der Vergleiche im schlimmsten Fall für dieses Verfahren und für das klassische Sortieren durch Verschmelzen nach dem Teile-und-herrsche-Prinzip:
 - i. Für Eingaben der Länge 2^k , für $k = 2, 3, 4, 5$.
 - ii. Für Eingaben der Länge $3 \cdot 2^{k-1}$, für $k = 2, 3, 4, 5$.
 - iii. Für Eingaben der Länge $2^{k+1} - 1$, für $k = 2, 3, 4, 5$.
- (b) Beweisen Sie, dass die Laufzeit dieses Verfahrens $O(n \log n)$ ist.

Diese Aufgabe ist für die wechselseitige Bewertung vorgesehen. Laden Sie die Lösung daher im KVV hoch, vorzugsweise als PDF-Datei oder ersatzweise als gescannte Bilddatei (*nicht* als veränderbare doc-, tex-, rtf- oder odt-Datei), und schreiben Sie Ihre Namen nicht in die Datei. Eine Abgabe auf Papier ist nicht notwendig.