

Übungsblatt 3

Abgabe bis Dienstag, den 13. Mai um 12:00 Uhr

Lassen Sie sich bei diesem Übungsblatt nicht von der Notation oder den langen Erklärungen abschrecken. Die Aufgaben sind nicht so schwer, wie es zuerst den Anschein haben mag. Man muss aber schon die Definition von "vergleichsbasiert" verstanden haben und was es mit den I/E-Sequenzen auf sich hat (nehmen Sie sich Zeit dafür und machen Sie es sich an konkreten Beispielen klar). Und man kann schön üben, was es bedeutet, eine Aussage wirklich zu beweisen.

Aufgabe 1 (10 Punkte)

Beweisen Sie, dass *MinSort* ein vergleichsbasierter Algorithmus ist. Argumentieren Sie dazu mit dem aktuellen Code aus der Vorlesung 3 (siehe Wiki). Beweisen Sie dazu die folgende Aussage (die in der Vorlesung ausführlich erklärt wurde): Es gibt eine Funktion f , die jede mögliche I/E-Sequenz von *MinSort* auf eine Permutation σ abbildet, so dass jede Eingabe, die diese I/E-Sequenz zur Folge hat, durch σ sortiert wird.

Hilfestellung: Machen Sie sich in Ruhe an einem oder mehreren Beispielen klar, wie die I/E-Sequenz die Permutation bestimmt, nach der *MinSort* die Eingabe schließlich sortiert. Sei die Eingabe zum Beispiel 17, 32, 4, 11. Dann ist die I/E-Sequenz des ersten Durchlaufs der äußeren Schleife genau **EEIEIEI**, wobei die fettgedruckten I/E vom Programmteil zum Update des Minimums stammen (I = neues Minimum, E = kein neues Minimum) und die nicht fettgedruckten vom Testen der Schleifenbedingung (I = Schleife beendet, E = Schleife läuft weiter, das erste E kommt vom Testen der Schleifenbedingung der äußeren Schleife). Das Ergebnis dieses Durchlaufs ist, dass die Elemente an den Stellen 1 und 3 der Eingabe (dort, wo 17 und 4 stehen) getauscht werden. Das entspricht mathematisch der Transposition $\tau_1 = (1\ 3)$. Eine Transposition ist eine Permutation, bei der genau zwei Stellen getauscht werden. Machen Sie sich klar, wie genau τ_1 mit der I/E-Sequenz zusammenhängt. Die weiteren Durchläufe der äußeren Schleife liefern ebenfalls Transpositionen $\tau_2, \tau_3, \dots$, die aus den I/E-Sequenzen folgen. Überlegen Sie, wie diese Transpositionen die Permutation σ ergeben, die die Eingabe korrekt sortiert.

Aufgabe 2 (10 Punkte)

In der Vorlesung 3 haben wir zusammen *ZeroOneSort* implementiert. Ändern Sie die Implementierung so ab, dass alle *konditionalen* Teile mit *if* und *else* realisiert sind, so wie für das *MinSort* aus Vorlesung 3. Denken Sie unbedingt daran, auch das Berechnen der Summe der Elemente (die *sum* Funktion) explizit auszuprogrammieren sowie das Erzeugen des Ergebnisfelds am Ende (damit die konditionalen Teile sichtbar werden). Der Algorithmus sollte für Eingabegröße n weiterhin $\leq C \cdot n$ Operationen benötigen, für eine Konstante C .

Beweisen Sie dann anhand eines Beispiels, dass Ihre Implementierung nicht vergleichsbasiert ist. Verstehen Sie, dass damit die lineare Laufzeit von *ZeroOneSort* nicht der unteren Schranke von $n \cdot \log n$ aus der Vorlesung widerspricht.

Hilfestellung: Finden Sie zwei Eingabefolgen I_1 und I_2 mit folgenden Eigenschaften: Erstens, es gibt keine Permutation, die gleichzeitig I_1 und I_2 sortiert. Zweitens, die I/E-Sequenz für ihre Implementierung ist für beide Eingabefolgen genau gleich. Machen Sie sich klar, warum das bedeutet, dass Ihre Implementierung nicht vergleichsbasiert ist.

Committen Sie Ihre Abgabe als ein PDF (bitte Regel 2 vom Wiki beachten) in das SVN, in einem Unterordner *blatt-03*. Auch Ihr Code für Aufgabe 2 sollte Teil dieses PDFs sein (es hilft bei der Argumentation, wenn man den Code dazu im selben Dokument sieht). Committen Sie in diesem Unterordner außerdem wie gehabt eine Textdatei *erfahrungen.txt*, in der Sie kurz Ihre Erfahrung mit diesem Übungsblatt und der Vorlesung dazu beschreiben, insbesondere wie lange Sie gebraucht haben und ob es wesentliche Probleme gab und wenn ja welche.

Von etwa 200 Studienanfänger:innen des B.Sc. Informatik schließen nur etwa 30 Personen das Studium auch ab. Woran, denken Sie, liegt das und wie passt das zu der Statistik von Folie 4 der Vorlesung 3?