

Prof. Dr. Martin Skutella
Torsten Gellert
Martin Groß
Dr. Max Klimm

Aylin Acikel, Katharina Bütow, Christian Döblin,
Alexander Hopp, Daniel Kuske, Olivia Röhrig
Robert Rudow, Daniel Schmand, Hendrik Schrezenmaier,
Judith Simon, Sebastian Spies, Fabian Wegscheider, Jan Zur

1. Übungsblatt Computerorientierte Mathematik II

Abgabe: 16.04.2013 (bis 14:15 Uhr in MA 004)

0. Aufgabe

(−5 bis 0 Punkte)

(a) Diese Aufgabe ist bis spätestens **Montag, den 15.04.2013, 16:00 Uhr**, von allen Teilnehmern der CoMa zu bearbeiten, die einen Übungsschein erhalten wollen!

Legt in eurem Gruppenaccount im UNIX-Pool eine Datei namens `members.txt` an, (z.B. `/homes/co2/co2-101/members.txt`) die die Daten Eurer Gruppenmitglieder in folgendem Format enthält. Für jedes Gruppenmitglied steht eine Zeile der Form:

`Vorname;Nachname;Matrikelnr.;Geschlecht;Studiengang;Emailadresse;tubitlogin;`

Sonstige Kriterien an den Inhalt der Datei:

- Ein Leerzeichen zwischen mehreren Vorname/Nachnamen ist erlaubt.
- Zusätzliche Leerzeichen, Leerzeilen oder sonstige Abweichungen vom geforderten Format sind nicht erlaubt.
- Wer aus keine Matrikelnummer hat, trägt an dieser Stelle **000000** ein.
- Wer keinen Tubit-Account hat, trägt an dieser Stelle **keintubit** ein.
- Schülerstudenten tragen als Studiengang **Schülerstudent** ein. Gasthörer, die für keinen Studiengang an einer Universität, Hochschule oder Fachhochschule immatrikuliert sind, tragen als Studiengang **Gasthörer** ein.
- Umlaute, Akzente usw. sollen in \LaTeX -Schreibweise geschrieben werden.
- Der Studiengang ist vollständig auszuschreiben. (Achtung, steht später auch so auf deinem Schein!)

Ein Beispiel für eine komplette Datei:

```
Lieschen;M"uller;123456;f;Wirtschaftsmathematik;liesel@gmx.net;lmueLLer;  
H"anschen;Klein;654321;m;Mathematik (Diplom);hansi.klein@web.de;hklein;  
Andr\'e;Strau"s;000000;m;Sch"ulerstudent;andre_strauss@mail.ru;keintubit;
```

Da wir diese Datei automatisch auswerten, müssen Name und Format genau den Vorgaben entsprechen. Beachtet die Kleinschreibung des Dateinamens `members.txt`.

Fehlt die Datei am 15.04.2013, 16:00 Uhr oder entspricht nicht den Vorgaben: -5 Punkte

(b) Legt außerdem in eurem Gruppenaccount im UNIX-Pool eine Datei namens `.forward` an (z.B. `/homes/co2/co2-101/.forward`) und schreibt eure Email-Adressen (eine pro Zeile) hinein. Dadurch werden auf eurem Account eingehende Emails an diese Adressen weitergeleitet. Nicht wundern, diese Datei ist durch den Punkt automatisch versteckt.

1. Aufgabe

(0 + 1 + 3 + 3 + 2 Punkte)

Ein $m \times n$ -Young Tableau ist eine $m \times n$ -Matrix, in der die Einträge (aus $\mathbb{R} \cup \{\infty\}$) jeder Zeile von links nach rechts monoton steigen und in jeder Spalte von oben nach unten monoton steigen. Einen Eintrag mit ∞ , nennen wir *leer*.

- (a) Gebt ein 4×4 -Young Tableau, mit den nicht-leeren Einträgen 10, 21, 3, 1, 4, 8, 5, 14, 12 an.
- (b) Zeigt, dass alle Einträge eines Young Tableaus Y leer sind, wenn $Y(1, 1)$ leer ist. Zeigt, dass alle Einträge eines $m \times n$ -Tableaus nicht leer sind, wenn $Y(m, n)$ nicht leer ist.
- (c) Entwerft einen Algorithmus mit Laufzeit in $\mathcal{O}(n+m)$, der aus einem gegebenen Young Tableau Y ein minimales Element löscht (d.h. durch ∞ ersetzt) und anschließend die Monotonie-Eigenschaften des Tableaus wieder herstellt.
- (d) Entwerft einen entsprechenden Algorithmus für das Einfügen eines Elements mit Laufzeit in $\mathcal{O}(n+m)$ in ein nicht volles Tableau (ein ∞ -Eintrag wird dabei durch das neue Element ersetzt).
- (e) Zeigt, wie man mit Hilfe eines Young Tableaus ohne Verwendung weiterer Sortierverfahren ein Array der Länge n^2 in $\mathcal{O}(n^3)$ sortieren kann.

2. Aufgabe

(3 + 3 Punkte)

Nehmt an, dass die Wahl des Pivot-Elements im Quicksort-Algorithmus durch einen Unteralgorithmus geschieht, der uns das Pivot-Element so aus dem Array der Größe n auswählt, dass nach dem Verschieben des Pivotelements genau r Elemente links des Pivotelements stehen. Einen solchen Unteralgorithmus nennen wir *Orakel* und ignorieren seine Funktionsweise und seinen Beitrag zum Sortieraufwand des Quicksort-Algorithmus.

- (a) Welchen Aufwand (in Θ -Notation) hat der beschriebene Quicksort-Algorithmus (ohne Berücksichtigung des Aufwandes für den Unteralgorithmus), wenn uns das Orakel immer das 2. Element – d. h. es gilt $r = 1$ – als Pivotelement gibt?
- (b) Welchen Aufwand (in Θ -Notation) hat der beschriebene Quicksort-Algorithmus (ohne Berücksichtigung des Aufwandes für den Unteralgorithmus), wenn uns das Orakel immer ein Pivotelement mit $r = \frac{11}{12}n$ gibt?

3. Aufgabe

(3 + 1 + 1 Punkte)

Gegeben ist ein Array der Länge n , das nur ganze Zahlen aus $\{0, 1, \dots, 9\}$ enthält.

- (a) Entwerft einen Algorithmus, der die Zahlen mit einem Aufwand von $\mathcal{O}(n)$ sortiert und beweist seine Korrektheit. Es dürfen keine Listen verwendet und keine Listenoperationen auf Arrays simuliert werden! (Zuweisungen an Laufvariablen dürfen vernachlässigt werden.)
- (b) Beweist den Aufwand eures Algorithmus aus (a) in Anzahl Zuweisungen (solche an Laufvariablen nicht mitgezählt).
- (c) Begründet, warum die Laufzeit $\mathcal{O}(n)$ eures Algorithmus nicht im Widerspruch zur unteren Sortierschranke steht.