Prof. Dr. Benjamin Doerr, Dr. Reto Spöhel
Übungsleitung: Alexander Kobel

Wintersemester 2011/12

Übung zu Grundzüge von Algorithmen und Datenstrukturen

http://www.mpi-inf.mpg.de/departments/d1/teaching/ws11/grads/

Übungsblatt 5 Abgabe: Donnerstag, 24. November 2011, 12:00 Uhr

Hinweise zur Abgabe der Aufgaben: Schreiben Sie klar und deutlich in gebührendem Abstand zu weiterem Text Ihren **Namen** und **Matrikelnummer**, ihre **Übungsgruppe als arabische Ziffer** sowie den **Namen Ihres Tutors** auf Ihre Abgabe. **Heften** Sie mehrere Blätter geeignet zusammen. Legen Sie ihre Lösung bis zum Abgabetermin in den **unteren rechten Briefkasten** im Erdgeschoss von Gebäude E1 3 (neben Hörsaal 001). Achtung: Verwechseln Sie den Briefkasten nicht mit dem für die *Stammvorlesung* Algorithms and Data Structures!

Aufgabe 0 (*Aufwärmübung, trotzdem schriftlich, 1 Punkt*)

Kern der letzten Vorlesung war ein Algorithmus, der ein Element i-ter Ordnung in einem unsortierten Feld findet. Lösen Sie nun das umgekehrte Problem: Geben Sie einen Algorithmus an, der zu einem unsortierten Feld a und einem Element x die Ordnung des Elementes im Feld berechnet (d.h. ein i, so dass x ein Element i-ter Ordnung in a ist, oder -1, falls x nicht in a enthalten ist). Geben Sie den Algorithmus in Pseudocode an. Keine weiteren Erklärungen oder Beweise nötig.

Aufgabe 1 (*Schriftliche Übung, 3 Punkte*)

a) Sei $q \neq 1$. Sei $s_n := \sum_{i=0}^{n-1} q^i$ für alle $n \in \mathbb{N}$. Finden Sie eine rekursive Beziehung, die für die $s_n, n \in \mathbb{N}$, gilt. Nutzen Sie diese und Lemma 1.10, um $s_n = (q^n - 1)/(q - 1)$ für alle $n \in \mathbb{N}$ zu zeigen.

b) Wann konvergiert die Reihe $\sum_{i=0}^{\infty} q^i$? Was ist ihr Grenzwert ℓ_q?

c) Sei $0 < q < 1$. Sei $t : \mathbb{N} \to \mathbb{R}_{>0}$, so dass $t(0) = 0$ und $t(n) \leq t(\lfloor qn \rfloor) + c\,n$ für ein $c > 0$ und alle $n \geq 1$ gilt. Zeigen Sie per Induktion, dass $t(n) \leq c\,\ell_q n$ für alle $n \in \mathbb{N}$ gilt.

Bitte wenden...

Aufgabe 2 (*Schriftliche Übung, 7 Punkte*)

a) Im Beweis der Anzahl der Vergleiche, die der median-of-medians-Algorithmus durchführt, wurde die sehr großzügige Abschätzung genutzt, dass Merge-Sort 5 Elemente mit maximal 12 Vergleichen sortiert. Wenn man die genauere rekursive Formel aus Skript/Vorlesung verwendet, sieht man, dass bereits 8 Vergleiche genügen. Nutzen Sie diese Abschätzung, um eine bessere Schranke für die Zahl der benötigten Vergleiche zu erhalten. Formulieren Sie eine Aussage analog zu (4.1) ohne Beweis und beweisen von da ab ihre Schlüsse. [2P.]

b) Analysieren Sie den median-of-medians-Algorithmus, der nicht 5er-Gruppen bildet, sondern 7er-Gruppen. Nutzen Sie, dass der Median von 7 Elementen mit 14 Vergleichen gefunden werden kann. Beweisen Sie sauber eine entsprechende Version von Lemma 4.4, von da ab genügen Skizzen der Rechnungen. [3P.]

c) Was geschieht, wenn wir 3er-Gruppen bilden statt 5er-Gruppen? Äussern Sie sich, ohne Beweis, zu Korrektheit und Laufzeit. [2P.]

Aufgabe 3 (*Schriftliche Übung, 3 Punkte*) Der in der Vorlesung vorgestellt Selektionsalgorithmus verwendet, dass alle Feldelemente verschieden sind. Modifizieren Sie den Algorithmus geeignet, so dass diese Einschränkung nicht mehr besteht.

Hinweis: Problematisch sind identische Feldelemente nur in Zeilen 5 und 6 des Algorithmus, d.h., in der Partitionsprozedur. Ändern Sie diesen Teil des Algorithmus geeignet. Beschreiben Sie zunächst informell Ihre Lösungsidee. Geben Sie dann Ihre Lösung in Pseudocode an, d.h., in einer präziseren Weise als im Skript. Kein Beweis notwendig.

Aufgabe 4 (*Schriftliche Übung, 2 Punkte*) Entwickeln Sie einen Algorithmus, um sowohl Minimum und Maximum eines n-elementigen Feldes zu finden. Ihr Methode soll weniger Vergleiche benötigen als der triviale Ansatz ($(n-1)+(n-2)$ Vergleiche). Wie viele Vergleiche benötigt Ihr Algorithmus? Eine informelle Beschreibung der Vorgehensweise genügt.

Bonusaufgabe (*4 wirklich einfache Bonuspunkte*)

Ersetzen Sie im Algorithmus Select aus Vorlesung/Skript die Zeilen 2-6 durch folgende Zeilen:

2: Wähle $i \in \{1, \ldots, n\}$ zufällig.
3: Setze $m := a[i]$.
4: Partitioniere a mit Pivot m (wie bisher in Zeile 5).
5: Sei q die Position von m in a.
6: Falls $q \notin [\frac{3}{10}n - 2, \frac{7}{10}n + 3]$, gehe zu Schritt 2.

Argumentieren Sie, warum der resultierende Algorithmus korrekt ist [1P.]. Zeigen Sie, dass die erwartete Anzahl von Vergleichen, bis die Schleife 2–6 verlassen wird, höchstens $2.5n$ ist [1P.; wenn Sie das nicht schaffen, können Sie den Rest dennoch lösen]. Geben Sie eine rekursive Beschreibung der erwarteten Anzahl der Vergleiche des Algorithmus auf einem Feld der Länge n an. Stören Sie sich nicht an den Erwartungswerten — der Erwartungswert ist linear und alles geht gut [1P.]. Lösen Sie die Rekursion und zeigen Sie somit eine obere Schranke für die durchschnittliche Zahl der Vergleiche [1P.].