Lehrstuhl für Algorith.
und Datenstrukturen
Prof. Dr. Hannah Bast
Claudius Korzen

Algorithmen und Datenstrukturen (Informatik II) SS 2015

http://ad-wiki.informatik.uni-freiburg.de/teaching

UNI FREIBURG

Übungsblatt 1

Abgabe bis Dienstag, den 28. April um 12:00 Uhr

Aufgabe 1 (5 Punkte)

Melden Sie sich bei unserem Kurssystem Daphne an. Den Link dazu finden Sie auf dem Wiki zum Kurs. Achten Sie darauf, dass Ihre Daten korrekt sind, insbesondere dass Sie unter der angegebenen E-Mail Adresse auch erreichbar sind.

Aufgabe 2 (5 Punkte)

Implementieren Sie das in der Vorlesung erklärte *QuickSort*. Beachten Sie dazu die auf dem Wiki verlinkte (programmiersprachen-unabhängige) Design-Vorlage *QuickSort.TIP*.

Grundsätzliche Bemerkungen zu den Implementierungen, siehe Seite 2 (gültig für alle Übungsblätter).

Aufgabe 3 (5 Punkte)

Messen Sie die Laufzeit von *QuickSort* für verschiedene Eingabegrößen n. Wählen Sie Ihre Eingaben alle so, dass die Elemente gerade verkehrt herum sortiert sind. Betrachten Sie zwei Arten, das Pivot-Element zu wählen: *fix* (erste Position) und *zufällig*.

Erstellen Sie für jede Art ein Schaubild, wie in der Vorlesung für *MinSort* gezeigt. Messen Sie dabei für den fixen Pivot für $n = 1.000, 1.400, \ldots, 5.000$ (5 Messungen für jedes n). Messen Sie für zufälligen Pivot für $n = 1.000, 2.000, \ldots, 20.000$ (ebenfalls 5 Messungen für jedes n).

Diskutieren Sie Ihre Ergebnisse kurz in Ihren *erfahrungen.txt* (siehe Aufgabe 4).

Aufgabe 4 (5 Punkte)

Committen Sie Ihren Code (samt Unit Tests) und die Schaubilder in das SVN, in einen eigenen Unterordner *uebungsblatt-01*. Beachten Sie dabei die Hinweise auf Seite 2.

Committen Sie in diesem Unterordner außerdem eine Textdatei *erfahrungen.txt*. Beschreiben Sie dort in ein paar Sätzen Ihre Erfahrungen mit diesem Übungsblatt und den Vorlesungen dazu. Insbesondere: Wie lange haben Sie ungefähr gebraucht? An welchen Stellen gab es Probleme und wieviel Zeit hat Sie das gekostet?

[bitte wenden[1]]

[1] Das Übungsblatt sollte von beiden Seiten gut angebraten werden, aber innen zart bleiben.

Grundsätzliche Bemerkungen zu den Implementierungen (gültig für alle Übungsblätter)

1. Als Programmiersprache stehen Ihnen grundsätzlich Python, Java und C++ zur freien Auswahl. Sie müssen sich auch nicht festlegen, sondern können sich für jedes Übungsblatt neu entscheiden.

2. Die Designvorlage (.TIP Datei, falls vorhanden) ist nicht verbindlich. Sie beruht aber auf viel Erfahrung, von daher sollten Sie sich dreimal überlegen, wenn Sie davon abweichen.

3. Schreiben Sie Ihren Code modular: wenn ein Teil des Codes für sich alleine umfangreich bzw. komplex genug ist oder mehrfach benötigt wird, gehört er in eine eigene Funktion.

4. Schreiben Sie für jede nicht-triviale Funktion einen Unit Test. Jeder Unit Test sollte mindestens ein nicht-triviales Beispiel überprüfen. Wenn es kritische Grenzfälle gibt, die sich durch wenig Aufwand leicht nachprüfen lassen (z.B. Verhalten einer Methode bei leerem Eingabefeld), sollten Sie das ebenfalls tun.

5. Laden Sie Ihren Code vollständig in unser SVN hoch, inklusive *Makefile* (für Python und C++) bzw. *build.xml* (Java). Andere Dateien (insbesondere *.pyc* oder *.o* oder *.class*) sollen nicht mit hochgeladen werden.

6. Die finale Version von Ihrem Code soll fehlerfrei auf Jenkins durchlaufen. Zwischenversionen (die Sie zum Beispiel rein zu Sicherungszwecken in unser SVN hochgeladen haben) müssen nicht auf Jenkins durchlaufen.

7. Wenn Sie ein Problem bei der Implementierung haben, suchen / googeln Sie ein paar Minuten nach dem Fehler, aber nicht zu lang. Fragen Sie dann lieber im Forum, da wird Ihnen in der Regel schnell geholfen. Eine Anleitung für das richtige Fragen auf dem Forum findet sich auf dem Wiki (Stichwort: konkret fragen mit Fehlermeldung + Zeilennummer + relevantem Code, und nicht nur „Mein Code geht nicht“).