



Algorithmen und Datenstrukturen

Sommersemester 2022

Musterlösung Übungsblatt 1

Abgabe: Dienstag, 3. Mai, 2022, 10:00 Uhr

Aufgabe 1: Anmeldung

(5 Punkte)

Melden Sie sich beim Kurssystem Daphne an. Den Link dazu finden Sie auch auf der Kurs-Website. Achten Sie darauf, dass Ihre Daten korrekt sind, insbesondere dass Sie unter der angegebenen E-Mail Adresse auch erreichbar sind. Führen Sie ein `checkout` auf Ihr SVN-Repository durch.¹

Aufgabe 2: Quicksort

(5 Punkte)

Implementieren Sie den in der Vorlesung erklärten Algorithmus *QuickSort* mit *zwei Varianten* der Pivot-Wahl: “Element an erster Position”, “Element an zufälliger Position”. Verwenden Sie dazu die auf der Webseite verlinkte Design-Vorlage `QuickSort.py`. Schreiben Sie je einen Unit Test für die `quicksort_divide` und die `quicksort_recursive` Methode. Die Unit Tests sollten grundsätzlich mindestens ein nicht-triviales Beispiel überprüfen. Wenn es kritische Grenzfälle gibt, die sich leicht nachprüfen lassen (z.B. Verhalten einer Methode bei leerem Eingabefeld), sollen Sie dies tun.

Musterlösung

Siehe `Quicksort.py` im public Ordner oder auf der Website.

Aufgabe 3: Zeitmessungen

(5 Punkte)

Messen Sie die Laufzeit Ihrer *QuickSort*-Implementierung mit beiden Varianten der Pivotwahl für jeweils zwei unterschiedliche Eingabearten. Die Eingabearten sind Arrays $[n, n-1, \dots, 2, 1]$ und Arrays die mit n *zufälligen* Integers gefüllt sind. Wiederholen Sie dies für die Eingabegrößen $n \in \{100, 200, \dots, 5000\}$.² Stellen Sie für jede der vier Varianten (Eingabeart, Pivotwahl) die Entwicklung der Laufzeit über n graphisch dar.³ Diskutieren Sie Ihre Ergebnisse kurz in Ihren `erfahrungen.txt` (siehe Aufgabe 4).

Musterlösung

Abbildungen 1 und 2 stellen die Laufzeiten jeweils mit unterschiedlicher y -Skala graphisch dar. Wir machen folgende Beobachtungen: Quicksort hat einen superlinearen (quadratischen) Trend für deterministische Pivotwahl (erstes Element) und bei umgekehrt sortierter Liste. Dahingegen ist *Quicksort*

¹Ihr SVN-Repository wird bei der ersten Anmeldung bei Daphne automatisch angelegt. Die URL ist: <https://daphne.informatik.uni-freiburg.de/ss2022/AlgoDat/svn/ihr-rz-account-name>

²Eine Funktion für das Erzeugen der Arrays und Ausgabe der Messergebnisse ist in `QuickSort.py` bereitgestellt.

³Die Unterschiede der Laufzeiten der Algorithmen werden am deutlichsten wenn diese gemeinsam in einem einzigen Schaubild aufgetragen werden mit n auf der x-Achse und der Laufzeit der entsprechenden Variante auf der y-Achse. Für ein klareres Bild können Sie sowohl eine *lineare* als auch eine *logarithmische y*-Skala ausprobieren.

für alle anderen Varianten in denen eine Randomisierung des Pivots oder der Eingabe stattfindet, für große Eingaben wesentlich schneller als die oberen Varianten, genauer: $\Theta(n \log n)$ “mit hoher Wahrscheinlichkeit”, siehe Vorlesung Woche 2).

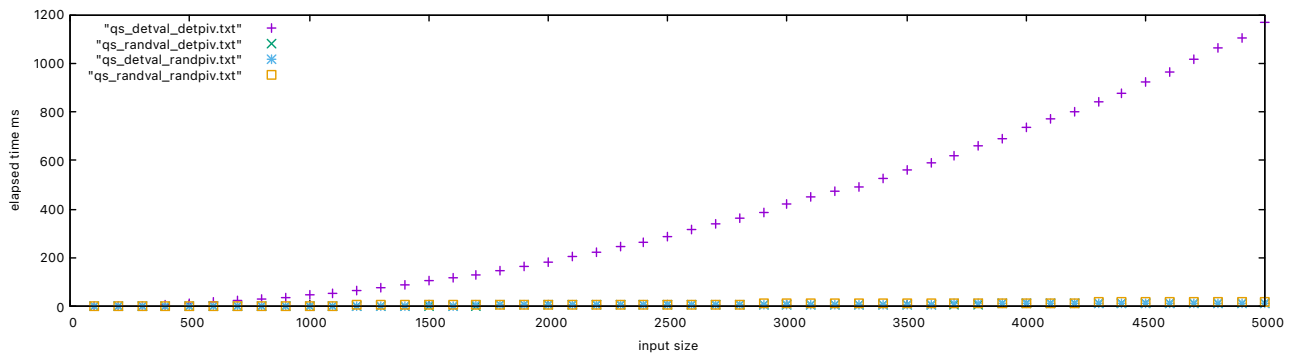


Abbildung 1: Der erste Plot gibt die Laufzeiten von allen geforderten Varianten von Sortieralgorithmen für die geforderten Eingaben über der Eingabegröße n an.

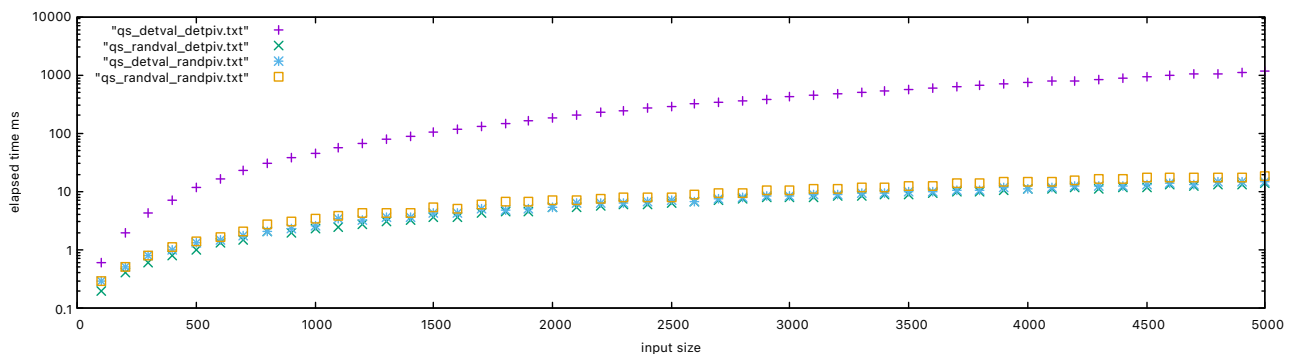


Abbildung 2: Der zweite Plot gibt die Laufzeiten von allen geforderten Varianten von Sortieralgorithmen für die geforderten Eingaben über der Eingabegröße n an. Die y -Skala ist logarithmisch.

Aufgabe 4: Abgabe

(5 Punkte)

Committen Sie Ihren Code (inkl. Tests) und die Schaubilder in das SVN, in einen eigenen Unterordner `uebungsblatt-01`. Gehen Sie dabei so vor, wie in der Vorlesung vorgeführt. Stellen Sie sicher, dass auf Jenkins alles (inkl. Style Check und Unit Tests) fehlerfrei durchläuft.

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?