

Übungsblatt 8

Abgabe bis Dienstag, den 27. Juni um 12:00 Uhr

Bei diesem Übungsblatt gibt es bewusst etwas weniger Vorgaben als sonst. Sie sollen ein Problem lösen und dabei selber entscheiden, welche der bisher kennengelernten Datenstrukturen dafür geeignet sind, und dies selbstverständlich auch gut (= mathematisch) begründen.

Das Problem ist wie folgt. Gegeben ist eine Liste mit Städten und ihren Ländern, Einwohnerzahlen und Geo-Koordinaten (siehe Wiki). Sie sollen diese Daten einlesen, abspeichern und dann in einer Endlosschleife Anfragen der folgenden Art beantworten:

Anfrage: ein Präfix (z.B. *frei*).

Antwort: alle Städte, die mit diesem Präfix anfangen (“case-insensitive”), und ihre zugehörigen Daten (Land + Einwohnerzahl + Geo-Koordinaten). Die Städte sollten nach Einwohnerzahl sortiert sein (höchste Einwohnerzahl zuerst).

Optional können Sie aus den Daten einen Link basteln, mit dem man auf den entsprechenden Ort auf einer der vielen Online-Karten (z.B. Google Maps oder OpenStreetMap) kommt.

Aufgabe 1 (10 Punkte)

Überlegen Sie sich, welche Datenstruktur für das beschriebene Problem geeignet sind. Begründen Sie Ihre Wahl, und zwar stichhaltig und nicht nur qualitativ sondern quantitativ. Schreiben Sie *keinen* Roman sondern wenige Sätze.

Bestimmen Sie außerdem (nachdem Sie Aufgabe 2 gemacht haben) die Laufzeit von Ihrem Programm und zwar sowohl die Laufzeit von dem Teil, in dem die Daten eingelesen werden und die Datenstruktur aufgebaut wird, als auch die Laufzeit für die Beantwortung einer Anfrage. Sie sollten die Laufzeit wie üblich als $\Theta(\dots)$ angeben. Entscheiden Sie selber, welche Variablen hier eine Rolle spielen.

Wichtig: Der Aufschrieb für diese Aufgabe sollte Teil von Ihrem Programm sein und ganz zuoberst (in der Hauptdatei, wenn es mehrere Dateien gibt) in einem Kommentar stehen.

Aufgabe 2 (10 Punkte)

Schreiben Sie ein Programm für das oben beschriebene Problem. Wenden Sie alles an, was Sie bisher gelernt haben und vermeiden Sie alle Fehler, die Sie inzwischen verlernt haben bzw. spätestens mit diesem Übungsblatt verlernt haben sollten.

Sie müssen die Datenstrukturen, die Sie verwenden, für diese Aufgabe nicht selber implementieren, sondern können einfach auf die von der jeweiligen Sprache zur Verfügung gestellten Datenstrukturen zurückgreifen.

Achten Sie wie gehabt auf die Hinweise auf der Rückseite vom ersten Übungsblatt. Dokumentieren Sie insbesondere Ihren Code ordentlich (kurz und präzise, keine Romane schreiben) und denken Sie an die Unit Tests und den Style und dass auf Jenkins alles fehlerfrei durchläuft.

Aufgabe 3 (optional, 5 Bonuspunkte)

In der Vorlesung 8b haben wir gezeigt, dass für einen $(2, 4)$ -Baum die Laufzeit für n Operationen (insert oder remove, in beliebiger Abfolge) $O(n)$ sind. Wir haben den Beweis mittels einer Potenzialfunktion Φ geführt, deren Wert für einen gegebenen Zustand des Baumes einfach die Anzahl der Knoten vom Grad 3 ist. Nehmen wir nun an, wir haben einen $(3, 7)$ -Baum und möchten den analogen Beweis führen. Welche Potenzialfunktion ist dann geeignet?

Diese Aufgabe gibt 5 Bonuspunkte. Die maximale Punktzahl für das Übungsblatt bleibt aber 20 Punkte. Das heißt, wenn Sie zusammen mit den Bonuspunkten auf über 20 Punkte kommen, bekommen Sie 20 Punkte für das Blatt.

Committen Sie Ihren Code (inklusive ihres Aufschrieb für Aufgabe 1, siehe oben) in unser SVN, in einen neuen Unterordner *blatt-08*, nach den bekannten Regeln der Kunst. Committen Sie in diesem Unterordner außerdem wie gehabt eine Textdatei *erfahrungen.txt*. Wer die Datei *cities.txt* mit in das SVN hochlädt, bekommt für jeden Einwohner einen Punkt Abzug.

Was war die entscheidende Mutation, die zum Homo Sapiens geführt hat und wann war das?