Lehrstuhl für Algorith.
und Datenstrukturen
Prof. Dr. Hannah Bast
Björn Buchhold

Algorithmen und Datenstrukturen (ESE) WS 2011 / 2012

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

UNI FREIBURG

Übungsblatt 4

Abgabe bis Dienstag, den 29. November um 16:00 Uhr

Aufgabe 1 (10 Punkte)

Implementieren Sie eine Klasse *HashMap*, die eine hash map, wie in der Vorlesung erklärt implementiert, mit ganzen Zahlen als Schlüsseln (key) und beliebigen Objekten als Wert (value).

Sie finden ein Gerüst der Klasse mit einer Spezifikation der zu implementierenden Funktionen (*insert*, *lookup*, *hash*) in den auf dem Wiki verlinkten Dateien zur Vorlesung. Die Hashfunktion soll im Konstruktor der Klasse zufällig gewählt werden. Wählen sie dazu eine der drei in der Vorlesung unter "Positivbeispiele" vorgestellten Klasse von Funktionen, welche ist ganz Ihnen überlassen. Schreiben Sie selbstverständlich und wie immer gute Unit Tests für alle Ihre Methoden und für den Konstruktor.

Aufgabe 2 (6 Punkte)

Implementieren Sie ein Programm *HashMapMain*, dass eine große Anzahl von Schlüsseln (keys) mit beliebigen Werten (values) in Ihre HashMap einfügt, und dann für eine gegebene Position in der Hashtabelle die Anzahl der Schlüssel ausgibt, die unter dieser Position gespeichert werden. Fügen Sie dazu Ihrer Klasse eine weitere Methode hinzu, die genau diese Anzahl zurückgibt.

Überprüfen Sie schließlich, dass die ausgebenen Zahl in Einklang mit der Universalität der von Ihnen in Aufgabe 1 ausgewählten Klasse ist. Das heißt, wenn Sie n Schlüssel in einer Hashtabelle mit m Positionen abgespeichert haben, und Ihre Klasse von Hashfunktionen c-universell ist, sollte die durchschnittliche Anzahl von Schlüsseln die auf eine feste(!) Position abgebildet werden $\leq c \cdot n/m$ sein. Wie genau sie n und m wählen und welche Position, ist Ihnen überlassen, aber sorgen Sie dafür, dass $n >> m$. Ein Beispiel wären $n = 1.000.000$, $m = 10.000$ und Position 0.

Wählen Sie *keine* zufällige Schlüsselmenge, sondern eine möglichst regelmäßige (und wenn Sie möchten, machen Sie das Experiment für mehrere Schlüsselmengen). Der Punkt von universellem Hashing ist ja gerade, dass es für *jede* Schlüsselmenge funktioniert. Für zufällig verteilte Schlüssel tun es ja auch ganz einfache Hashfunktionen wie x mod m.

Aufgabe 2 (4 Punkte)

Committen Sie Ihre Lösung in das SVN, und den Code in einem neuen Unterordner *src/uebungsblatt_4*. Schauen Sie, dass auf Jenkins (unserem Build-System, siehe Link auf Ihrer Daphne Seite) alles ohne Fehler durchläuft.

Schreiben und committen Sie außerdem wie gehabt (in *non-code/uebungsblatt_4*) eine Datei *erfahrungen.txt* zu Ihren Erfahrungen mit dem Übungsblatt und der Vorlesung.