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

Übungsblatt 5

Abgabe bis Dienstag, den 2. Juni um 12:00 Uhr

Aufgabe 1 (5 Punkte)

In der Vorlesung haben wir 5 Klassen von Hashfunktionen besprochen (zwei Negativbeispiele, drei Positivbeispiele). Schreiben Sie für jede dieser Klassen eine Funktion, die für ein gegebenes Schlüsselpaar x, y die Kollisionswahrscheinlichkeit $\Pr(h(x) = h(y))$ berechnet. Sie müssen dazu über alle Hashfunktionen der jeweiligen Klasse iterieren. Wenn das zu lange dauert, können Sie alternativ einen Schätzwert für die Kollisionswahrscheinlichkeit berechnen. Berechnen Sie dazu $\hat{p}(x, y, N) = |\{h_i : h_i(x) = h_i(y)\}|/N$, wobei $h_1, \ldots, h_N$ zufällig und unabhängig voneinander aus der Klasse gewählt sind. Wählen Sie N dabei so groß, dass die Laufzeit noch erträglich ist (wenige Sekunden).

Aufgabe 2 (5 Punkte)

Erstellen Sie für jede der fünf Klassen aus Aufgabe 1 ein Histogramm der Kollisionswahrscheinlichkeiten für alle Paare von Schlüsseln x, y mit $x \neq y$. Wählen Sie $U = \{0, ..., 255\}$ als Schlüsseluniversum und $m = 16$ für die Größe der Hashtabelle (oder $m = 17$, wenn die Größe eine Primzahl sein muss). Was ein Histogramm ist, wurde in der Vorlesung erklärt.

Geben Sie außerdem für jeder der fünf Klassen die Laufzeit zur Erstellung des Histogrammes an (als $\Theta(...)$, in Abhängigkeit von $|U|$ und m).

Aufgabe 3 (5 Punkte)

Erklären Sie für jedes Histogramm, inwiefern man daraus ersehen kann, ob die betreffende Klasse von Hashfunktionen universell ist oder nicht, und wenn ja mit welchem Wert für c.

Zusatzaufgabe (5 Punkte): Manche der Histogramme sind „multimodal“, d.h. sie haben mehrere „Spitzen“. Geben Sie für jede dieser Spitzen eine plausible Erklärung, insbesondere für die Höhe und warum es gerade bei dieser Kollisionswahrscheinlichkeit eine „Spitze“ gibt.

[bitte schwungvoll wenden]

Aufgabe 4 (5 Punkte)

Entwerfen und implementieren Sie einen Algorithmus, der für eine gegebene Mindestgröße m einer Hashtabelle die kleinste Primzahl p mit $p \geq m$ findet. Sie können dabei das Betrandsche Postulat benutzen, wonach es für alle $m \geq 3$ eine Primzahl p zwischen m und $2m$ gibt. Bestimmen Sie die Laufzeit von Ihrem Algorithmus (2 Punkte).

Zusatzaufgabe (5 Punkte): Versuchen Sie, so nah wie möglich an lineare Laufzeit heranzukommen. Behaupten Sie die bessere Laufzeit nicht einfach nur, sondern begründen Sie sie.

Committen Sie Ihren Code, sowie die Histogramme und die dazugehörigen Erklärungen (bitte alles in einem PDF, liebevoll verpackt) in unser SVN, in einen neuen Unterordner *uebungsblatt-05*. Die Coding Standards sind Ihnen inzwischen in Fleisch und Blut übergegangen und Sie können gar nicht mehr anders.

Committen Sie in diesem Unterordner außerdem wie gehabt eine Textdatei *erfahrungen.txt*. Beschreiben Sie dort, wie gehabt, in ein paar Sätzen Ihre Erfahrungen mit diesem Übungsblatt und den Vorlesungen dazu.