

Übungsblatt 9

Abgabe für ESE: bis Dienstag, den 8. Januar um 16:00 Uhr

Abgabe für IEMS: bis Dienstag, den 22. Januar um 16:00 Uhr

Aufgabe 1 (5 Punkte)

Implementieren Sie eine Klasse *StaticHashSet* mit den zwei Methoden *fill* und *lookup*. Spezifikation und Design Vorschlag siehe Wiki. Wählen Sie als Hashfunktion einfach $h(x) = x \bmod m$, wobei m die Größe der Hashtabelle ist.

Aufgabe 2 (5 Punkte)

Schreiben Sie einen Unit Test für die beiden Methoden aus Aufgabe 1. Das ist hier besonders wichtig, weil es bei Ihren Optimierungen in Aufgabe 4 unten schnell passieren kann, dass Sie zwar die Laufzeit beschleunigen, die Implementierung aber nicht mehr korrekt ist. Dem beugen Unit Tests vor.

Aufgabe 3 (5 Punkte)

Schreiben Sie ein Programm *HashSetMain*, das $n = 10$ Millionen zufällige positive ganze Zahlen erzeugt und diese in eine Hashtabelle der Größe n/C einfügt. Schauen Sie selber, welcher Wert von $C \geq 1$ Ihnen die besten Resultate bringt. Verwenden Sie dazu einmal Ihre Klasse aus Aufgabe 1 und einmal *HashSet<Integer>* (Java) bzw. `__gnu_cxx::hash_set<int>` oder `std::unordered_set<int>` (C++). Wiederholen Sie die Ausführung für jede der beiden Varianten mindestens drei Mal, und vergleichen Sie die Laufzeiten.

Aufgabe 4 (5 Punkte)

Versuchen Sie Ihre Implementierung aus Aufgabe 1 so zu verbessern, dass die Laufzeit aus Aufgabe 3 möglichst klein wird. Fassen Sie in Ihren *erfahrungen.txt* kurz zusammen, welche Änderungen Sie ausprobiert haben, ob Sie was gebracht haben, und wenn ja wie viel. Tragen Sie Ihr bestes Ergebnis in die auf dem Wiki verlinkte Tabelle ein.

Committen Sie wie gehabt Ihren Code in das SVN, in einen neuen Unterordner *uebungsblatt_09*, und ebenso Ihr Feedback in einer Textdatei *erfahrungen.txt*. Insbesondere: Wie lange haben Sie ungefähr gebraucht? An welchen Stellen gab es Probleme und wieviel Zeit hat Sie das gekostet?