**Assignment 8**

**Mouna Kacimi, Werner Nutt, Camilo Thorne**

# Hashing

## 1. Optimizing Chaining

Instead of using lists for storing entries with the same hash values, it is also possible to use binary search trees.

1. What are in this case the worst-case running times for insert, find, and delete? Do they improve?
2. What are in this case the average running times for insert, find, and delete? Do they improve?

(6 Points)

## 2. Counting Words by Hashing

Counting words in a text is a basic task in linguistic analysis. For example in the first two lines of this exercise, there are 29 words, of which 24 are different.

1. Describe how you can effectively count the number of different words in a text using a hash table.
2. Implement a hash table of your choice (with open addressing or chaining) with $2^{14}$ cells that can use any of the following three Java hash functions for words:
   - `String.hashcode();`
   - `String.substring(0,4).hashcode();`
   - a hash function chosen by yourself, that may only use the characters of the word (you may use the function `Character.hashcode()`).

3. Apply your hashtable to each of following three texts:

   - the book of Genesis[1];
   - Romeo and Juliet by Shakespeare[2];
   - the European regulation about banana quality standards[3].

   For each text, count how many different words are in the text. Also count the percentage of different words wrt. the total number of words in each text (e.g., 6,000 different words in 8,000 words in total gives a percentage of 75%).

4. For each text, count the number of collisions for each hash function. What differences did you find? How can they be explained? How well did each of the three functions perform?

**Hints:**

- It is advisable to copy the texts from Wikisource into textfiles, then read the textfiles linewise in Java (using classes such as BufferedReader, InputStreamReader or similar).
- To identify the words in the text, you may use the Java class StringTokenizer.
- Ignore sentence delimiters (“.”, “,”, “\”, “!”, “?”, “;”), numbers and other special characters.
- Also ignore cases, you may use the function `String.toLowerCase()` for that purpose.

(24 Points)

Submission: Until Thu, 28 May 2014, 8:30 pm, to

`dsa-submissions AT inf DOT unibz DOT it.`

[1] `http://en.wikisource.org/wiki/Bible_%28King_James%29/Genesis`
[2] `http://en.wikisource.org/wiki/The_Tragedy_of_Romeo_and_Juliet`
[3] `http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31994R2257:EN:HTML`