

Woche 11: Hashing

Aufgabe 1: Map mit Separate Chaining

In der Vorlesung haben Sie verschiedene Arten kennengelernt, wie Sie mit Kollisionen beim Hashen umgehen können. Implementieren Sie eine Hashmap, die für die Kollisionsbehandlung *separate chaining* verwendet. Dafür werden bei einer Kollision die **Entries** als Liste miteinander verbunden. Dafür können Sie die **Entry**-Klasse verwenden.

```
public class Entry<K, V> {  
    private final K key;  
    private V value;  
    private Entry<K, V> next;  
    ...  
}
```

Aufgabe 2: Vergleich von Hash-Funktionen

Ihre Aufgabe ist es verschiedene Hashfunktionen für den Datentyp String zu schreiben und diese zu vergleichen.

Implementieren Sie folgende Hash-Funktionen:

1. Integer-Cast
2. Komponentensumme
3. Polynom-Akkumulation
4. Eigene Hash-Funktion

Ihre Hashfunktionen sollten diesen Funktions-Körper haben:

```
public static int hashCode(String s) {  
  
}
```

Sie können dann in der Klasse **HashFunctionEvaluator** die Hashfunktion evaluieren. Diese Funktion erzeugt alle möglichen Strings bis zu einer max. Länge und wendet ihre Hashfunktion, die Sie als Parameter übergeben können darauf an, um alle Kollisionen festzustellen.

In der Klasse **Visualizer** Können Sie die Kollisionen als LineChart darstellen.

Merkmale guter Hashfunktionen:

- kurze, konstante Laufzeit
- geben für den Wert **x** immer denselben Hash zurück (sonst ist es keine Valide Hash-Funktion)
- es sollten möglichst wenig Kollisionen vorhanden sein und diese sollten gleichmässig verteilt sein

Aufgabe 3: Set und Map

Die Klasse **WordIndex** soll in der Lage sein eine Text-Datei zu indexieren. Implementieren Sie hierzu die Funktionen *indexFile* und *getLines*. Die Datei soll so indexiert sein, dass wenn nach einem Wort gesucht wird die Zeilen in

welchen das Word vorkommt zurückgegeben werden. Verwenden Sie für die Implementation Maps und Sets.

```
void indexFile(String path)
Integer[] getLines(String word)
```