Wilhelm BuchmÃ¼ller (3133783)
Daniel Wanner (TODO)
Artur Frenzen (TODO)

# Aufgabe 1: Textalgorithmen/Levenshtein-Distanz

(a)

| | $\epsilon$ | t | r | e | k | k | i | e | s |
|---|---|---|---|---|---|---|---|---|---|
| $\epsilon$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| w | 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| a | 2 | 2 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| r | 3 | 3 | 5 | 3 | 4 | 5 | 6 | 7 | 8 |
| s | 4 | 4 | 3 | 3 | 4 | 5 | 6 | 7 | 7 |

Abbildung 1: Die Levenshtein-Distanz-Tabelle zu trekkies und wars

(b) Die Levenshtein-Distanz zwischen zwei Worten befindet sich immer in der untersten rechten Ecke.

# Aufgabe 2: Hashfunktionen

(a) Das englische Alphabet hat nur 26 Buchstaben, d.h das englische Wort kann nur mit 26 Buchstaben anfangen, der Buchstabe des Wortanfangs ich auch nicht gleichverteilt. Man kann also nicht von den vollen 128 Werten von UTF-8 profitieren. Die Surjektivität ist gegeben, da max. 26 Buchstaben benutzt werden, und nicht 128. Auch die Gleichverteiltheit ist nicht gegeben, da die Worte mit Abstand am häufigsten mit T und am seltensten mit X beginnen.

(b) Es gibt 50 reservierte keywords in der Java Programmiersprache, 53 wenn man `true`, `false` und `null` dazurechnet und 27 Operatoren, wenn man new und instanceof zu den keywords zählt. Die Eigenschaft der Surjektivität ist gegeben es gibt keywords/Operatoren von Länge 1 bis Länge 6. Auch die Eigenschaft der Gleichverteiltheit ist gegeben.

(c) Die Hashfunktion ist nicht surjektiv, da nicht jeder Student der an der Vorlesung auch zwingend an der U"bung teilnimmt, auch ist es nicht sehr wahrscheinlich das jede Punktzahl von z.B. 1/X bis X/X einmal erreicht wird. Da es außerdem eine Mindestpunktzahl (für den Schein) von 60 % gibt werden sich alle Studenten anstrengen den Schein zu bestehen und demnach ist die Gleichverteiltheit entsprechend davon beeinflußt, also nicht erfüllt.

(d) Die Surjektivität ist gegeben, da auf alle Ziffern der Hashfunktion abgebildet wird. Die Gleichverteilheit ist auch gegeben, da von 0-`INTEGER.MAX` alle Werte gleich verteilt werden.

Wilhelm BuchmÃ¼ller (3133783)
Daniel Wanner (TODO)
Artur Frenzen (TODO)

# Aufgabe 3: Hashing

(a)

| Index | Entry | | | |
|---|---|---|---|---|
| 0 | | | | |
| 1 | | | | |
| 2 | | | | |
| 3 | 16 | 68 | 94 | 3 |
| 4 | 82 | | | |
| 5 | | | | |
| 6 | | | | |
| 7 | | | | |
| 8 | | | | |
| 9 | | | | |
| 10 | | | | |
| 11 | | | | |
| 12 | | | | |

Abbildung 2: Die offene Hashtabelle hat 3 Kollisionen an Index 3

(b)

| Index | Entry |
|---|---|
| 0 | |
| 1 | |
| 2 | |
| 3 | 16 |
| 4 | 82 |
| 5 | 68 |
| 6 | |
| 7 | 94 |
| 8 | |
| 9 | 3 |
| 10 | |
| 11 | |
| 12 | |

Abbildung 3: geschlossene Hashtabelle mit linearem Sondieren (keine Kollisionen)

Wilhelm BuchmÃ¼ller (3133783)
Daniel Wanner (TODO)
Artur Frenzen (TODO)

(c)

| Index | Entry |
|---|---|
| 0 | |
| 1 | |
| 2 | |
| 3 | 16 |
| 4 | 82 |
| 5 | 68 |
| 6 | |
| 7 | 94 |
| 8 | |
| 9 | 3 |
| 10 | |
| 11 | |
| 12 | |

Abbildung 4: geschlossene Hashtabelle mit quadriertem Sondieren (keine Kollisionen)