

Aufgabe 1: Textalgorithmen/Levenshtein-Distanz

(a)

	ε	t	r	e	k	k	i	e	s
ε	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 ist 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 Gleichverteilung 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 Gleichverteilung ist gegeben.
- (c) Die Hashfunktion ist surjektiv, da jeder Student eine Gesamtpunktzahl hat - auch Studenten die abbrechen oder nichts abgeben erreichen rechnerisch 0 Gesamtpunkte. Da es eine Mindestpunktzahl (für den Schein) von 60 % werden sich alle Studenten anstrengen den Schein zu bestehen und demnach ist die Gleichverteilung entsprechend davon beeinflusst.
- (d) Die Surjektivität ist gegeben, da auf alle Ziffern der Hashfunktion abgebildet wird. Die Gleichverteilung ist auch gegeben, da von 0-`INTEGER.MAX` alle Werte gleich verteilt werden.

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 Hashtabelle hat 3 Kollisionen an Index 3

(b) PLATZHALTER

(c) PLATZHALTER