Institut für Informatik
Dr. Tobias Neckel
Dr. Andreas Paul
Benjamin Peherstorfer

Technische Universität München
WS 2013/2014

Einführung in die wissenschaftliche Programmierung

Übungsblatt 11

1.) LZ77 Komprimierung

LZ77 ist ein Algorithmus zur verlustfreien Datenkompression. Dieser Algorithmus macht sich zunutze, dass ganze Wörter in einem Text oft mehrfach vorkommen. Dazu verwendet man einen Vorschau-Puffer und ein Lexikon fester Größe. Im Vorschau-Puffer befindet sich immer der gerade aktuelle Teil des noch zu komprimierenden Texts. Bei jedem Schritt prüft man nun ob eine (möglichst lange) Zeichenfolge im Vorschau-Puffer bereits im Lexikon vorkommt. Wenn ja, dann wird ein Tripel (Codewort) codiert. Der erste Eintrag ist die Position im Lexikon, der zweite Eintrag die Länge des zu kopierenden Teils und der dritten Eintrag das folgende Zeichen. Findet man keine brauchbare Zeichenfolge, wird ein Tripel der Form (0, 0, 'x') codiert, wobei 'x' das folgende Zeichen im Vorschau-Puffer ist. Anschließend muss das Lexikon aktualisiert werden, d.h. die komprimierten Zeichen wandern in das Lexikon. Die Funktionsweise soll anhand der Komprimierung von BANANE dargestellt werden (mit Vorschau-Puffer- und Lexikongröße 4):

i) Lexikon ist leer, d.h. es kann nichts verwendet werden ⇒ (0, 0, B)

ii) Vorschau-Puffer enthält keine Zeichenfolge des Lexikons ⇒ (0, 0, A)

iii) Wieder keine brauchbare Zeichenfolge im Lexikon ⇒ (0, 0, N)

iv) Jetzt findet man AN im Lexikon und im Vorschau-Puffer. AN ist an der Position 1 und es sollen 2 Zeichen kopiert werden, anschließend folgt ein E ⇒ (1, 2, E)

Lexikon				Vorschaupuffer				Codewörter
				B	A	N	A	
				0	1	2	3	
			B	A	N	A	N	(0, 0, B)
			0	0	1	2	3	
		B	A	N	A	N	E	(0, 0, A)
		0	1	0	1	2	3	
	B	A	N	A	N	E		(0, 0, N)
	0	1	2	0	1	2		
N	A	N	E					(1, 2, E)
0	1	2	3					

Beim Dekomprimieren muss das Lexikon wieder aufgebaut werden um Zeichenfolgen zu kopieren. Das Dekomprimieren von BANANE:

i) Nichts zu kopieren, hänge Zeichen B an und aktualisiere Lexikon ⇒ B

ii) Nichts zu kopieren, hänge Zeichen A an und aktualisiere Lexikon ⇒ BA

iii) Nichts zu kopieren, hänge Zeichen N an und aktualisiere Lexikon ⇒ BAN

iv) Kopiere ab Position 1 genau 2 Zeichen und hänge E an ⇒ BANANE

Klartext	Lexikon				Codewörter
				B	(0, 0, B)
				0	
			B	A	(0, 0, A)
			0	1	
		B	A	N	(0, 0, N)
		0	1	2	
BA	N	A	N	E	(1, 2, E)
	0	1	2	3	

Implementieren Sie den LZ77 Algorithmus zum Komprimieren und Dekomprimieren in Python. Schreiben Sie zusätzlich zwei Funktionen `compress` und `decompress` welche eine Eingabe- und eine Ausgabedatei und die Vorschau-Puffer (z.B. 128) und Lexikon Größe (z.B. 4096) als Parameter bekommen.

Es gibt viele Möglichkeiten die Codewörter in eine Datei zu schreiben. Sie können z.B. das Modul `pickle` verwenden um ganze Datenstrukturen (z.B. Listen) in eine Datei zu schreiben:

```
import pickle
l = [1, 2, 3, 4, 5]

# Liste in Datei schreiben
fp = open("mylist.txt", "wb")
pickle.dump(l, fp)
fp.close()

# Liste aus Datei lesen
fp = open("mylist.txt", "rb")
ll = pickle.load(fp)
fp.close()
```

Vergleichen Sie Ihre Version mit bekannten anderen Komprimierungsprogrammen.

Optional:
Eine speichersparendere Methode kann mit dem Modul `struct` implementiert werden, siehe `http://docs.python.org/library/struct.html`.