Buchstaben-Bäume

Schreiben Sie ein Programm, welches eine **Liste von Wörtern** (z.B. die in `http://www.cis.uni-muenchen.de/~schmid/lehre/data/wortliste`) aus einer Datei einliest und in einen Buchstabenbaum (Trie) umwandelt. Jede Kante von einem Knoten zu einem Tochterknoten ist mit einem Buchstaben beschriftet. Es ist nicht erlaubt, dass 2 (ausgehende) Kanten eines Knotens denselben Buchstaben tragen. Jeder Pfad vom Wurzelknoten des Baumes zu einem Terminalknoten soll einem Wort entsprechen und umgekehrt. Dies ist jedoch nicht gewährleistet, wenn ein Wort Präfix eines anderen Wortes sein kann. Daher sollen Terminalknoten explizit markiert werden, so dass auch Knoten mit Übergängen zu anderen Knoten Terminalknoten sein können.

Aufruf: `python build-tree.py wordlist.txt > tree.txt`

Der Buchstabenbaum soll als Liste von Kanten auf den Bildschirm ausgegeben werden. Jede Ausgabezeile enthält eine Kante bestehend aus dem Index des Startknotens, dem Buchstaben und dem Index des Zielknotens. Als Trennzeichen sollen Tabulatoren verwendet werden. Der Startknoten hat den Index 0. Am Ende geben Sie noch die Liste der Terminalknoten aus, indem Sie den Index jedes Terminalknotens auf einer separaten Zeile ausgeben.

Beispiel: Ein Trie für das Wort “ab”:

```
0	a	1
1	b	2
2
```

Schreiben Sie nun noch ein Programm, welches den Buchstabenbaum aus einer Datei einliest und dann Wörter Zeile für Zeile aus einer zweiten Datei einliest und prüft, ob sie im Buchstabenbaum (und damit in der ursprünglichen Wortliste) enthalten sind.

Aufruf: `python lookup.py tree.txt words.txt`

Jedes Eingabewort wird in einer separaten Zeile zusammen mit der Bewertung `known` oder `unknown` ausgegeben.

Vorüberlegungen

- Erstellen Sie von Hand den Baum für die Wortliste: reitet reicht riecht
- Welche Datenstruktur verwenden Sie für den Baum?
- Wie repräsentieren Sie die Information über Terminalknoten?