Prof. für Algorithmen
und Datenstrukturen
Prof. Dr. Hannah Bast
Dr. Patrick Brosi

Algorithmen und Datenstrukturen
SS 2025

https://ad-wiki.informatik.uni-freiburg.de/teaching

UNI FREIBURG

Übungsblatt 8

Abgabe bis Dienstag, den 01. Juli um 12:00 Uhr

Aufgabe 1 (10 Punkte)

Implementieren Sie für einen binären Suchbaum die folgenden beiden Methoden. Folgen Sie dabei der Vorlage auf dem Wiki, die bereits eine Methode *insert(key, value)* enthält.

lookup(low, upp): Rückgabe aller Elemente, deren Schlüssel mindestens *low* und höchstens *upp* ist. Die Elemente sollten aufsteigend nach ihrem Schlüssel sortiert sein. Die Laufzeit dieser Funktion sollte $O(d+k)$ sein, wobei d die Tiefe des Baumes und k die Anzahl der Element im Ergebnis ist.

erase(key): Löschen des gegebenen Schlüssels. Wenn es den Schlüssel nicht im Baum gibt, soll nichts passieren. Die Laufzeit dieser Funktion sollte $O(d)$ sein, wobei d die Tiefe des Baumes ist.

Aufgabe 2 (10 Punkte)

Auf dem Wiki finden Sie einen Datensatz mit einer Liste von Adressen und ihrem Link auf das entsprechende Objekt in OpenStreetMap. Ihr Code soll diese Daten einlesen und in einem binären Suchbaum gemäß Aufgabe 1 abspeichern. Sie können dabei annehmen, dass die Adressen eindeutig sind. Dann soll Ihre Anwendung wie folgt auf Nutzereingaben reagieren:

<*prefix*>: Ausgabe aller Adressen, deren Name mit <*prefix*> beginnt (ohne Unterscheidung zwischen Groß- und Kleinschreibung) mit ihrem Link auf das Objekt in OpenStreetMap. Die Adressen sollen dabei alphabetisch sortiert sein.

!insert <*place*>;<*osm_link*>: Einfügen der gegebenen Adresse mit dem gegebenen Link. Falls es <*place*> bereits gibt, soll der entsprechende Eintrag überschrieben werden.

!erase <*place*>: Löschen dieser Adresse. Falls es <*place*> nicht gibt, soll nichts passieren.

In der Codevorlage ist bereits einiger Code vorgegeben, um Ihnen etwas Arbeit abzunehmen, insbesondere Code für das Einlesen der Daten und die Anfrageschleife. Das Einlesen der n Adressen sollte in Zeit $O(n \cdot \log n)$ laufen, *insert* und *erase* wie in Aufgabe 1.

[bitte umblättern, auf der nächsten Seite stehen noch wich]

Auf dem Wiki sind drei Dateien verlinkt: *places-freiburg.tsv* (34 k Adressen), *places-bawue.tsv* (2.8 M Adressen) und *places-germany.tsv* (19 M Adressen). Ihr Programm sollte mindestens mit der ersten davon funktionieren, aber probieren Sie gerne auch die anderen.

Wenn Sie Ihre Anwendung so wie bisher beschrieben implementieren, reicht das zwar für die volle Punktzahl für die Aufgabe, die Suche ist aber nur beschränkt nützlich. Mit ein paar einfachen Erweiterungen können Sie die Suche deutlich komfortabler machen. Hier drei Vorschläge dazu und vielleicht haben Sie ja auch noch eigene Ideen:

1. Wenn nach Eingabe eines Präfixes *Return* gedrückt wird, erscheint nicht die vollständige Liste aller Treffer, sondern nur die Top-5 sowie ein Hinweis, falls es mehr Treffer gibt und wie viele. Dann erscheint wieder der Eingabeprompt, aber der bisherige Präfix ist schon voreingetragen, so dass man einfach von da aus weiter tippen (oder ihn auch wieder löschen) kann. Das geht einfach mit Pythons *readline* Modul.

2. Nehmen wir an, der bisherige Präfix ist *Stock*, alle bisherigen Treffer haben aber den Präfix *Stockmattenweg* gemeinsam. Dann kann man den voreingetragenen Präfix gleich auf *Stockmattenweg* erweitern. Das spart Tipparbeit. Da die Treffer sortiert sind, reicht es, sich dazu den ersten und letzten Treffer anzuschauen.

3. Mit dem Python-Modul *prompt_toolkit* kann man es auch noch interaktiver machen. Dann kann mit der Tabulator-Taste der Präfix verlängert werden, ohne dass man die Zeile mit dem Prompt wiederholen muss. Wenn keine Verlängerung möglich ist, kann man einfach wieder die Top-5 anzeigen, so wie unter 1. beschrieben. Also ganz ähnlich, wie sich die Tabulator-Taste in einer Konsole verhält, in der Autovervollständigung aktiviert ist.

Committen Sie Ihren Code und Ihre *erfahrungen.txt* in einen neuen Unterordner *blatt-08*.

Wie hoch rangiert die aktuelle geopolitische Lage auf der Liste Ihrer Sorgen?