Albert-Ludwigs-Universität
Institut für Informatik
Prof. Dr. F. Kuhn
P. Bamberger
P. Schneider

Algorithmen und Datenstrukturen
Sommersemester 2020
Übungsblatt 6

Abgabe: Mittwoch, 24.06.2020, 16:00 Uhr.

Aufgabe 1: Binärer Suchbaum - Range Queries *(10 Punkte)*

(a) Implementieren Sie die Datenstruktur des binären Suchbaumes (BST) und die `insert` Operation. Sie können dazu die Vorlage `BST.py` benutzen. *(4 Punkte)*

(b) Erweitern Sie Ihren BST um die Operation `getrange`(x_{min}, x_{max}) welche alle Schlüssel $x \in$ BST mit $x_{min} \leq x < x_{max}$ ausgibt. *(4 Punkte)*

Bemerkung: Ähnlich zu, aber nicht exakt gleich wie in Vorlesung Woche 6 Folie 21.

(c) Wir können auch Wörter über den Zeichen $\{a, \ldots, z\}$ bezüglich der *lexikographischen Ordnung*[1] in einen Binärbaum einfügen. Tun Sie dies für alle Wörter in der gegebenen Datei `inputs.txt`. Benutzen Sie diese Datenstruktur um alle Wörter mit einem bestimmten Präfix *effizient* auszugeben. Führen Sie als Unit-Test eine Suche nach allen Wörtern mit dem Präfix "qw" durch. Kopieren Sie die Ausgabe in Ihre `erfahrungen.txt`. *(2 Punkte)*

Aufgabe 2: Binärer Suchbaum - Operationen *(10 Punkte)*

(a) Beschreiben Sie eine Funktion, welche die Tiefe eines binären Suchbaums ausgibt, und analysieren Sie die Laufzeit. *(2 Punkte)*

(b) Beschreiben Sie eine Funktion, welche für einen gegebenen binären Suchbaum mit n Knoten und ein gegebenes $k \leq n$ eine Liste mit den k kleinsten Schlüsseln ausgibt. Analysieren Sie die Laufzeit in Abhängigkeit von k und der Tiefe d des Baumes. *(4 Punkte)*

(c) Beschreiben Sie eine Funktion, welche als Eingabe einen binären Suchbaum B und einen Schlüssel x erhält und folgende Ausgabe generiert:

- Falls es ein Element v in B gibt mit $v.key = x$, gebe v zurück.
- Andernfalls gebe ein Paar (u, w) zurück, wobei u das Baum-Element mit dem nächstkleineren und w das Element mit dem nächstgrößeren Schlüssel ist. Dabei soll $u =$ None falls x kleiner als alle Schlüssel im Baum ist und $w =$ None falls x größer als alle Schlüssel im Baum ist.

Die Beschreibung der Funktion kann in Pseudo-Code oder durch eine hinreichend genaue textliche Beschreibung erfolgen. Analysieren Sie die Laufzeit Ihrer Funktion. *(4 Punkte)*

[1] Python unterstützt Vergleiche von Wörtern bzgl. der lexikographischen Ordnung nativ.