Algorithmen und Datenstrukturen
Übungszettel 6

Martin Avanzini <martin.avanzini@uibk.ac.at>
Thomas Bauereiß <thomas.bauereiss@uibk.ac.at>
Herbert Jordan <herbert@dps.uibk.ac.at>
René Thiemann <rene.thiemann@uibk.ac.at>

3. Mai, zur Besprechung am 10. Mai

Aufgabe 1) Splay-Bäume

1. Führe folgende Operationen auf einem anfangs leerem Splay-Baum aus:

   E4,E7,E13,E15,E1,E4,L4,L15,E4,S12,S7

   Hier bezeichnet E Einfügen (put), L Löschen (remove) und S Suchen (get).

2. In Splay-Bäumen ist die Suchbaumeigenschaft "links < Schlüssel < rechts" immer gewährleistet.
   (a) Begründen Sie kurz, warum `splay(..)` die Suchbaumeigenschaft erhält.
   (b) Begründen Sie, warum beim Einfügen die Suchbaumeigenschaft erhalten bleibt.

Aufgabe 2) Hashing

1. Was wird als Kollision bezeichnet?
2. Erläutere die zwei grundlegenden Methoden zur Kollisionsauflösung.
3. Führe folgende Operationen auf einer anfangs leeren Hashtabelle der Größe 11 aus:

   E0,E23,E4,E6,E8,E10,E12,E1,E3,L1,E5,E14

   Hier bezeichnet Ek Einfügen mit Schlüssel k und Lk Löschen eines Elementes mit Schlüssel. Zur Kollisionsauflösung verwende jeweils
   (a) lineares Sondieren, und
   (b) quadratisches Sondieren.
4. Vervollständige die diesem Zettel beiliegende Implementierung von Hashing mit offener Adressierung.

(a) In der Vorlesung wurde eine unvollständige Version von `void put(K key, D data)` vorgestellt. Vervollständige diese Methode für den Fall `status[pos] == REMOVED`. Beachte dass `put(K key, D data)` mehrfaches Vorkommen gleicher Schlüssel ausschliessen muss.

(b) Implementiere die Methode `void remove(K key)` zum Löschen von Elementen.