Probeklausur zur Vorlesung

Algorithmen, Datenstrukturen und Datenabstraktion WS 17/18

Wolfgang Mulzer, Katharina Klost

Bearbeitungszeit: mindestens 90 Minuten

Aufgabe 1 Implementierungen eines ADT *3+3+4 Punkte*

Betrachten Sie die folgende Spezifikation eines abstrakten Datentypen: Sei $\mathcal{U}$ ein total geordnetes Universum. Es sollen Teilmengen $S \subseteq U$ gespeichert werden, so dass folgende Operationen möglich sind:

- `insert`(x): Voraussetzung: keine. Wirkung: $S \mapsto S \cup \{x\}$.
- `deleteMin`(): Voraussetzung: S ist nicht leer. Wirkung: $S \mapsto S \setminus \{\min S\}$.
- `deleteMax`(): Voraussetzung: S ist nicht leer. Wirkung: $S \mapsto S \setminus \{\max S\}$.

Sie dürfen annehmen, dass sich zwei Elemente aus $\mathcal{U}$ in konstanter Zeit vergleichen lassen. Für jede der folgenden drei Datenstrukturen, beschreiben Sie jeweils kurz, wie man die Operationen `deleteMin` und `deleteMax` möglichst effizient implementieren kann, und geben Sie möglichst gute asymptotische obere Schranken für die Laufzeit. Erklären Sie gegebenenfalls, welche zusätzlichen Annahmen nötig sind.

(a) Sortierte doppelt verkettete Liste mit Zeigern auf das erste und das letzte Element;

(b) AVL-Baum;

(c) binärer Min-Heap.

Aufgabe 2 Hashing *2+4+4 Punkte*

(a) Hashing mit Verkettung bietet $O(1)$ *erwartete* Laufzeit pro Operation. Was bedeutet das? Was kann man über die *worst-case* Laufzeit sagen?

(b) Sei T eine Hashtabelle mit Verkettung, die m Plätze hat und eine Menge S von $2m$ Einträgen speichert. Zeigen Sie: Unter der Annahme, dass die Hashfunktion h zufällig gewählt ist, ist die erwartete Zeit zum Finden eines festen Elementes x in der Hashtabelle $O(1)$. Nehmen Sie an, dass $h(x)$ in $O(1)$ Zeit berechnet werden kann.

(c) Sei T eine Hashtabelle mit n Plätzen, in der wir eine Menge S mit n Einträgen speichern. Sei h eine Hashfunktion.

Was versteht man unter einer *Kollision* unter h? Berechnen Sie die erwartete Anzahl *aller* Kollisionen in T unter der Annahme, dass sich h wie eine zufällige Funktion verhält.

Hinweis: Betrachten Sie alle Paare von Einträgen aus S und verwenden Sie die Linearität des Erwartungswerts.

Aufgabe 3 Vermischtes *3+2+2+3 Punkte*

(a) Was versteht man unter einer kryptographischen Hashfunktion? Nennen Sie zwei Eigenschaften und eine Anwendung.

(b) Was ist der Unterschied zwischen dem dynamischen und dem statischen Datentypen einer Variable? Geben Sie ein kurzes Beispielfragment in Java.

(c) Unter welchen Umständen ist eine Datenstruktur, die $O(\log n)$ *amortisierte* Laufzeit für eine Operation benötigt, einer Datenstruktur vorzuziehen, die $O(\log n)$ *worst-case* Laufzeit für die gleiche Operation benötigt?

(d) Was versteht man unter einem *abstrakten Datentypen*? Nennen Sie zwei Beispiele aus der Vorlesung und erklären Sie das Prinzip, auf dem das Konzept des abstrakten Datentypen basiert.

Aufgabe 4 Skiplisten *4+3+3 Punkte*

(a) Beweisen Sie: Die erwartete Größe einer zufällig aufgebauten Skipliste mit n Einträgen ist $O(n)$.

(b) Seien L_1 und L_2 zwei zufällig aufgebaute Skiplisten, welche jeweils die Schlüsselmenge K_1 und K_2 speichern. Beschreiben Sie einen effizienten Algorithmus, der aus L_1 und L_2 eine Skipliste für die Menge $K_1 \cup K_2$ konstruiert. Analysieren Sie die erwartete Laufzeit Ihres Algorithmus. (Sie können dabei die Aussage von (a) verwenden.)

(c) Nehmen Sie nun bei (b) zusätzlich an, dass für alle $k_1 \in K_1$, $k_2 \in K_2$ gilt: $k_1 < k_2$. Das heißt, jeder Schlüssel in L_1 ist kleiner als alle Schlüssel in K_2. Geben Sie nun einen Algorithmus an, der L_1 und L_2 in erwarteter Zeit $O(\max\{\log |K_1|, \log |K_2|\})$ vereinigt, und beweisen Sie die Laufzeitgarantie.

Hinweis: Für alle $x \in (0, 1)$ gilt: $\sum_{i=1}^{\infty} ix^i = x/(1-x)^2$.