6. Aufgabenblatt zur Vorlesung

Algorithmen, Datenstrukturen und Datenabstraktion WS 17/18

Wolfgang Mulzer, Katharina Klost

Abgabe am 01. Dezember 2017 bis 10 Uhr in die jeweiligen Tutorenfächer

Aufgabe 1 Implementierung weiterer Heap-Operationen *10 Punkte*

(a) Programmieren Sie einen binären Heap mit n Elementen, gespeichert als Array. Implementieren Sie außerdem die folgenden Operationen mit jeweils einer Laufzeit von $O(\log n)$:

- `decreaseKey`(i, k): Setze das Element an Position i auf den Wert k und stelle die Heapeigenschaft wieder her (Vorbedingung: k ist kleiner gleich dem alten Element an Position i).
- `delete`(i): Lösche das Element an der Position i und stelle die Heapeigenschaft wieder her.

Hierbei dürfen `decreaseKey` und `delete` auf `bubble-up` und `bubble-down` zurückgreifen. Begründen Sie kurz Korrektheit und Laufzeit.

(b) Wir wollen das Interface `PriorityQueue` aus der Vorlesung um die Operationen aus (a) erweitern. Erklären Sie kurz, warum wir zur effizienten Implementierung einen "Zeiger" in die Datenstruktur benötigen (d.h., warum ist es keine gute Idee, an `decreaseKey` nur das alte und das neue Element zu übergeben, selbst wenn jedes Element nur einmal in der Datenstruktur vorkommt?).

Wie können wir solche Zeiger implementieren, so dass das Geheimnisprinzip gewahrt bleibt? Führen Sie ein Interface `Position` ein, das einen Zeiger in die Datenstruktur repräsentiert. Erläutern Sie knapp, wie man das Interface `PriorityQueue` modifizieren kann, um `Position` zu benutzen. Welche Schwierigkeiten ergeben sich dabei? Skizzieren Sie kurz einen Lösungsansatz. (Bei dieser Teilaufgabe müssen Sie nichts programmieren.)

Aufgabe 2 Elementare Wahrscheinlichkeitsrechnung *10 Punkte*

Auf einem Tisch stehen N Kisten. In diese Kisten werden nacheinander unabhängig voneinander n Bälle geworfen, wobei jede Kiste mit gleicher Wahrscheinlichkeit getroffen wird.

(a) Berechnen Sie die Wahrscheinlichkeit, dass Kiste i leer ist. Sei Y_i die Zufallsvariable, die den Wert 1 annimmt, falls Kiste i leer ist, und 0 sonst. Geben Sie auch den Erwartungswert $E[Y_i]$ an.

(b) Sei X die Zufallsvariable, welche die Anzahl von leeren Kisten angibt. Berechnen Sie den Erwartungswert von X mit Hilfe der Erwartungswerte $E[Y_i]$.

(c) Geben Sie eine möglichst gute Schranke $f(N)$ an, so dass gilt: wenn $n \geq f(N)$, dann ist die Wahrscheinlichkeit, dass mindestens eine Kiste mindestens zwei Bälle enthält, größer als 1/2.

Hinweis: Stellen Sie sich vor, die n Bälle werden nacheinander in die Kisten geworfen. Was ist die Wahrscheinlichkeit, dass der nächste Ball in einer leeren Kiste landet, wenn alle vorherigen Bälle in einer leeren Kiste gelandet sind? Verwenden Sie die ungemein nützliche Abschätzung $1+x \leq e^x$, welche für alle $x \in \mathbb{R}$ gilt.

Aufgabe 3 Hashing im Selbstversuch *10 Punkte*

(a) Fügen Sie nacheinander die Schlüssel 5, 28, 19, 15, 20, 33, 12, 17, 10 in eine Hashtabelle der Größe 9 ein. Die Hashfunktion sei $h(k) = k \bmod 9$. Die Konflikte werden mit Verkettung gelöst.

(b) Fügen Sie nacheinander die Schlüssel 10, 22, 31, 4, 15, 28, 17, 88, 59 in eine Hashtabelle der Größe 11 ein. Die Hashfunktion sei $h(k) = k \bmod 11$. Die Konflikte werden durch offene Adressierung mit linearem Sondieren gelöst.

(c) Fügen Sie nacheinander die Schlüssel 10, 22, 31, 4, 15, 29, 17, 88, 59 in eine Hashtabelle der Größe 11 ein. Die Konflikte werden durch Kuckuck gelöst, mit $h_1(k) = k \bmod 11$ und $h_2(k) = (k \bmod 13) \bmod 11$.

Illustrieren Sie jeweils die einzelnen Schritte.

Hinweis: Pseudocode für die Hashoperation findet sich auf der Website.

Hinweis: Bitte formatieren Sie Ihre Abgaben mit LaTeX oder einem vergleichbaren elektronischen Textverarbeitungssystem. Ansonsten droht der Abzug von bis zu 10% der erreichbaren Punkte.