

Departement Informatik
Markus Püschel David Steurer Peter Widmayer
Chih-Hung Liu

30. Oktober 2017

Algorithmen & Datenstrukturen**Blatt 6****HS 17**

Übungsstunde (Raum & Zeit): _____

Abgegeben von: _____

Korrigiert von: _____

erreichte Punkte: _____

Aufgabe 6.1 *Minima und Maxima finden.*

Gegeben sei ein Array $A = (A[1], \dots, A[n])$ ganzzahliger Schlüssel.

- a) Entwerfen Sie einen Algorithmus, der den kleinsten Schlüssel in A findet, und analysieren Sie die Anzahl der Schlüsselvergleiche im schlechtesten Fall (exakt, nicht in $\mathcal{O}$ -Notation).

Hinweis: Wundern Sie sich nicht, wenn Ihnen diese Aufgabe überaus einfach vorkommt.

- b) Entwerfen Sie nun einen Algorithmus, der sowohl den kleinsten als auch den grössten Schlüssel in A findet. Analysieren Sie erneut die exakte Anzahl der verglichenen Schlüssel im schlechtesten Fall.

Hinweis: Natürlich könnte man den Algorithmus aus a) zweimal anwenden, um jeweils das Minimum und das Maximum getrennt voneinander zu finden. Ihr Algorithmus sollte also deutlich weniger als doppelt so viele Schlüssel wie der Algorithmus aus a) vergleichen.

Aufgabe 6.2 *Münzen mit Balkenwaage wägen.*

Gegeben sei eine Menge von n Münzen. Diese enthält genau eine falsche Münze, die schwerer als alle anderen Münzen ist. Um die falsche Münze zu finden, dürfen Sie nur eine Balkenwaage benutzen. Mit dieser kann lediglich ermittelt werden, ob die Münzen in der linken Schale leichter, schwerer oder genauso schwer sind wie die Münzen in der rechten Schale.



- a) Entwerfen Sie eine Strategie für $n = 9$ Münzen, die möglichst wenige Wägungen ausführt.
- b) Geben Sie für ein allgemeines n einen Algorithmus an, der im schlechtesten Fall genau $\log_3(n)$ viele Wägungen benötigt.

Hinweis: Sie dürfen annehmen, dass n eine Potenz von 3 ist.

Aufgabe 6.3 *Erweiterte Heaps.*

Sie haben den Max-Heap in der Vorlesung kennengelernt und gesehen, wie daraus das Maximum effizient extrahiert werden kann. In dieser Aufgabe sollen Sie eine Datenstruktur entwerfen, bei der sowohl das Maximum als auch das Minimum effizient extrahiert werden kann.

Beschreiben Sie den genauen Aufbau der Datenstruktur und wie darauf die folgenden Operationen effizient (d.h., mit einer Laufzeit in $\mathcal{O}(\log n)$) implementiert werden können, wenn n die Anzahl Elemente in der Datenstruktur ist:

- (i) EXTRACT-MAX: Gibt den grössten Schlüssel in der Datenstruktur aus und entfernt ihn anschliessend.
- (ii) EXTRACT-MIN: Gibt den kleinsten Schlüssel in der Datenstruktur aus und entfernt ihn anschliessend.
- (iii) INSERT(k): Fügt den neuen Schlüssel mit Wert k in die Datenstruktur ein.

Nehmen Sie dazu an, dass die Schlüssel in einem Array A gespeichert sind und dass A gross genug dimensioniert wurde, um alle jemals auftretenden Schlüssel zu speichern. Wie kann die Datenstruktur in Zeit $\mathcal{O}(n)$ initialisiert werden, sodass sie alle Elemente aus A verwaltet?

Hinweis: Natürlich muss sichergestellt werden, dass nach der Ausführung jeder Operation auch die nächste Operation effizient möglich ist.

Aufgabe 6.4 *Zahlensummen.*

Gegeben seien ein Array $A = (A[1], \dots, A[n])$ ganzzahliger Schlüssel sowie eine weitere ganze Zahl z . Geben Sie für die folgenden beiden Probleme einen möglichst effizienten Algorithmus an, und analysieren Sie seine Laufzeit (in $\mathcal{O}$ -Notation) im schlechtesten Fall.

- a) Es soll entschieden werden, ob das Array zwei Schlüssel $A[i]$ und $A[j]$ enthält, deren Summe z ergibt.
- b) Es soll entschieden werden, ob das Array drei Schlüssel $A[i]$, $A[j]$ sowie $A[k]$ enthält, deren Summe z ergibt.

Abgabe: Am Montag, den 6. November 2017 zu Beginn Ihrer Übungsgruppe.