Eidgenössische
Technische Hochschule
Zürich

Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Federal Institute of Technology at Zurich

Departement Informatik
Markus Püschel
Peter Widmayer
Thomas Tschager
Tobias Pröger

20. Oktober 2016

Datenstrukturen & Algorithmen — Blatt 5 — HS 16

Abgabe: Am Donnerstag, den 27.10.2016, vor Beginn der Vorlesung um 10 Uhr im Eingangsbereich vor ML D28. Bitte heften Sie Ihre Blätter zusammen und benutzen Sie dieses Blatt als Deckblatt. Füllen Sie auch die ersten zwei der untenstehenden Felder aus.

Übungsstunde (Raum & Zeit): _____

Abgegeben von: _____

Korrigiert von: _____

erreichte Punkte: ____

Aufgabe 5.1 *Suche nach gleichen Zahlen.*

Gegeben seien zwei Arrays A und B, die jeweils m bzw. n paarweise verschiedene Zahlen speichern. Durch ein geeignetes Vergleichsverfahren soll nun festgestellt werden, ob es Zahlen gibt, die sowohl in A als auch in B vorhanden sind.

a) Beschreiben Sie ein naives Verfahren, das alle solchen Zahlenpaare in unsortierten Arrays findet, und analysieren Sie anschliessend den Zeitaufwand. Geben Sie dazu an, wieviele Vergleiche im schlechtesten Fall ausgeführt werden, abhängig von den Arraygrössen m und n.

b) Angenommen, die Arrays A und B sind sortiert. Beschreiben Sie ein Verfahren, welches das oben beschriebene Problem im schlechtesten Fall in Zeit $\mathcal{O}(m+n)$ löst.

Aufgabe 5.2 *Erweiterte Heaps.*

In dieser Aufgabe soll ein Array $A[1..|A|]$, das einen Min-Heap repräsentiert, zur Verwaltung einer Menge von n Schlüsseln eingesetzt werden. Beschreiben Sie, wie die folgenden Operationen effizient (d.h., mit einer Laufzeit in $\mathcal{O}(\log n)$) implementiert werden können.

a) MIN: Gibt den kleinsten Schlüssel aus.

b) REPLACE(i, k): Entfernt den Schlüssel $A[i]$ und ersetzt ihn durch k.

c) INSERT(k): Fügt den neuen Schlüssel mit Wert k in den Heap ein.

d) DELETE(i): Entfernt den Schlüssel $A[i]$ aus dem Heap.

Hinweis: Natürlich muss sichergestellt werden, dass nach der Ausführung jeder Operation wieder ein Heap vorliegt. Sie dürfen davon ausgehen, dass A gross genug dimensioniert wurde, um alle auftretenden Schlüssel zu speichern.

Aufgabe 5.3 *Suchen in sortierten Arrays.*

In der Vorlesung wurden sowohl die binäre als auch die Interpolationssuche vorgestellt. Während erstere immer mit $\mathcal{O}(\log n)$ Schritten auskommt, braucht letztere im Mittel nur $\mathcal{O}(\log \log n)$ viele Schritte, im schlimmsten Fall jedoch $\Theta(n)$ viele.

a) Geben Sie eine unendliche Familie von Beispielen an (je eines für jede Folgenlänge n), wo die Interpolationssuche schneller ist als die binäre Suche, und eine andere unendliche Familie von Beispielen an, wo die binäre Suche schneller ist als die Interpolationssuche.

b) Die Interpolationssuche hat in schlimmsten Fall eine Laufzeit von $\mathcal{O}(n)$, und im Mittel eine Laufzeit von $\mathcal{O}(\log(\log(n)))$. Wie kann man die worst-case-Laufzeit des binären Suchens von $\mathcal{O}(\log n)$ und die mittlere Laufzeit der Interpolationssuche gleichzeitig erreichen?