

Übungsblatt 4

Aufgabe 4.1

6 Punkte

Wir wollen die Datenstruktur *binärer Suchbaum* um die Funktion **Smallest** erweitern. Diese erhält als Parameter eine natürliche Zahl k und soll den k -t kleinsten Schlüssel des Baumes zurückliefern. Ist k größer als die Anzahl der Schlüssel in dem Baum, dann soll ein Fehler zurückgegeben werden. Dazu dürfen zusätzliche Informationen in den einzelnen Knoten gespeichert werden.

Implementieren Sie die **Smallest**-Funktion. Beschreiben Sie, welche Zusatzinformationen Sie in den Knoten speichern und wie diese beim Aufruf der **Insert**- und der **Delete**-Funktion zu aktualisieren sind.

Das Ausführen jeder der drei Funktionen **Smallest**, **Insert** und **Delete** soll in Zeit $O(h)$ möglich sein, wobei h die aktuelle Höhe des Suchbaumes ist.

Aufgabe 4.2

3+3 Punkte

Gegeben sei ein Feld der Größe $m = 13$, das als Hashtabelle mit den Indizes $0, \dots, m - 1$ dient. Fügen Sie die Schlüssel

1207, 286, 1195, 298, 62, 573, 623, 6280, 89, 199, 447

in der angegebenen Reihenfolge gemäß der Hashfunktion $h(x) := x \bmod m$ in die Hashtabelle ein. (Es genügt, das Endergebnis aller Einfügeoperationen anzugeben.) Ermitteln Sie die durchschnittliche und die maximale Anzahl von Sondierungen, die beim Einfügen eines Schlüssels nötig waren. Betrachten Sie als Strategien

- (a) Lineares Sondieren
- (b) Quadratisches Sondieren

Aufgabe 4.3

3+3+6 Punkte

Gegeben seien zwei Felder A und B der Länge m bzw. n mit jeweils paarweise verschiedenen natürlichen Zahlen als Einträgen. Es soll getestet werden, ob $\{A[1], \dots, A[m]\}$ eine Teilmenge von $\{B[1], \dots, B[n]\}$ ist.

- (a) Geben Sie einen deterministischen Algorithmus an, der unter der Voraussetzung, dass A und B aufsteigend sortiert sind, obiges Problem in Zeit $O(n)$ löst. Dabei darf $O(1)$ zusätzlicher Speicher verwendet werden.
- (b) Geben Sie einen deterministischen Algorithmus an, der obiges Problem in Zeit $O(n)$ löst. Dabei darf $O(\Delta)$ zusätzlicher Speicher verwendet werden, wobei Δ die Differenz zwischen dem größten und dem kleinsten Eintrag von B ist.
- (c) Geben Sie einen randomisierten Algorithmus an, der obiges Problem in erwarteter Zeit $O(n)$ löst. Dabei darf $O(n)$ zusätzlicher Speicher verwendet werden.

Begründen Sie jeweils, warum der von Ihnen angegebene Algorithmus die Laufzeit- und Speicherrestriktionen einhält. Beachten Sie, dass die Felder A und B in Teil (b) und (c) nicht sortiert sein müssen.

Hinweis: Nutzen Sie Wissen und Ideen aus der Vorlesung und der Übung.