technische universität dortmund

SoSe 2017

C. Sohler
A. Krivošija, A. Rey, H. Sandvoß

DAP2 – Heimübung 11

Ausgabedatum: 23. 6. 17 — Abgabedatum: Fr. 30. 6. 17 (Mo. 3. 7. für Gruppen 27–32) 12 Uhr

Schreiben Sie unbedingt immer Ihren **vollständigen Namen**, **Ihre Matrikelnummer** und **Ihre Gruppennummer** auf Ihre Abgaben!

Aufgabe 11.1 (4 Punkte): (Streaming-Algorithmen)

In der Vorlesung (Foliensatz 14, Folien 3 und 4) wird das Problem der fehlenden Zahl vorgestellt: Das Universum U ist gegeben als die Menge $\{1, \ldots, N\}$ der ersten N natürlichen Zahlen. Im Datenstrom kommt jedes Element des Universums genau ein mal vor, bis auf das gesuchte Element a, das im Datenstrom fehlt. Ein Streaming-Algorithmus soll dieses Element mit $\mathcal{O}(\log N)$ Speicherplatz finden.

Gelöst wird die Aufgabe, indem man alle Elemente des Datenstroms aufaddiert. Sei $S = (N(N+1)/2)$ die Summe der ersten N natürlichen Zahlen und X die Summe der Elemente, die im Datenstrom vorhanden sind. Ein Streaming-Algorithmus kann diese Summe X unter Verwendung lediglich einer Variable berechnen, indem er all diese Elemente aus dem Datenstrom aufaddiert. Das fehlende Element a lässt sich dann angeben als $a = S - X$. Die Zahl $X \in \mathcal{O}(N^2)$ kann mit $\mathcal{O}(\log N)$ Bits gespeichert werden, da $X < S \leq N^2$ und damit $\lceil \log X \rceil \leq \log S + 1 \leq 2 \log N + 1$ gilt.

Wir modifizieren das Problem nun und setzen voraus, dass *genau zwei* Elemente $a, b \in U$ im Datenstrom fehlen. Ihre Aufgabe ist es, mit einem Streaming-Algorithmus diese beiden Elemente zu finden und dabei die Platzschranke $\mathcal{O}(\log N)$ beizubehalten.

a) Überlegen Sie, warum es nun nicht mehr ausreicht, lediglich die Summe der Elemente aus dem Datenstrom aufzusummieren.

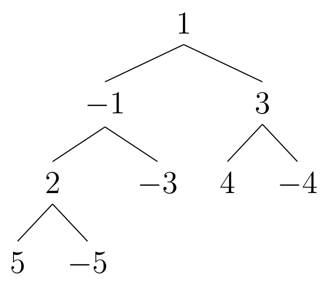
b) Beschreiben Sie, wie man mit Hilfe einer *zweiten* Variable das Problem lösen kann. Es muss *nicht* explizit ein Algorithmus in Pseudocode formuliert werden, für jeden verwendeten Wert muss jedoch klar sein, wie dieser berechnet werden kann.

c) Beweisen Sie, dass der von ihnen angegebene Algorithmus die Platzschranke $\mathcal{O}(\log N)$ beibehält.

Aufgabe 11.2 (6 Punkte + 4 Bonuspunkte): (Teile und Herrsche im Baum II)

Im Folgenden betrachten wir Binärbäume, deren Schlüsseleinträge unterschiedliche ganze Zahlen sind. In einem gegebenen binären Baum T gibt es für zwei Knoten a_0 und a_m in T einen eindeutigen Pfad $(a_0, a_1, \ldots, a_m)$ der Länge m von a_0 nach a_m (Die Länge ist also die Anzahl der Kanten). Wir betrachten solche Pfade, die aufsteigend sortiert sind, d. h., für deren Schlüssel $S(a_{i-1}) < S(a_i)$, $1 \leq i \leq m$, gilt. Wir suchen die Länge m eines längsten solchen Pfades.

a) (1 Punkt) Bestimmen Sie die Länge eines längsten aufsteigend sortierten Pfades im folgenden Baum und geben Sie die Schlüssel der zugehörigen Start- und Zielknoten an.



b) (5 Punkte) Entwerfen Sie einen Teile-und-Herrsche-Algorithmus, der bei Eingabe eines binären Baumes T die Länge eines längsten aufsteigend sortierten Pfades in T bestimmt, und beschreiben Sie ihn mit eigenen Worten. Geben Sie eine Implementierung Ihres Algorithmus in Pseudocode an.

c) (2 Bonuspunkte) Analysieren Sie die Laufzeit Ihres Algorithmus. Stellen Sie hierzu eine Rekursionsgleichung für die Laufzeit Ihres Algorithmus auf und lösen Sie diese.

d) (2 Bonuspunkte) Zeigen Sie die Korrektheit Ihres Algorithmus.