Karlsruher Institut für Technologie

Algorithmen 1 – Sommersemester 2018
Zuverlässige Softwaresysteme im Kontext der Automobilindustrie
`http://verialg.iti.kit.edu`

Prof. Dr. Carsten Sinz – Markus Iser

Übungsblatt 5

Ausgabe: 23.05.2018 – 15:30
Abgabe: 30.05.2018 – 13:00

A Algorithmendesign (2 Punkte)

Gegeben sei eine sortierte Liste mit n Elementen. Nun haben Sie $k \leq n$ neue unsortierte Elemente in einem Array, die in die sortierte Liste einfügt werden sollen, so dass die Liste nach dem Einfügen weiter sortiert ist.

1. *Warmup:* Geben Sie einen Algorithmus an, der Laufzeit $\mathcal{O}(k \cdot n)$ hat und das Problem löst.
2. Geben Sie nun einen Algorithmus an, der Laufzeit $\mathcal{O}(k \cdot \mathsf{log}(k) + n)$ hat und das Problem löst.

B Sortieren

Gegeben sei die Liste $A = \langle 3, 4, 5, 1, 17, 19, 13, 14 \rangle$.

B.1 Quantifizierung (1 Punkt)

Bestimmen Sie die Anzahl der Inversionen und Runs in A.

B.2 Sortieren mit Insertionsort (1 Punkt)

Sortieren Sie A mit der "In-Place" Variante von Insertionsort. Geben Sie den Zustand von A nach jedem Insert-Schritt an.

B.3 Sortieren mit Mergesort (1 Punkt)

Sortieren Sie A mit der Variante von Mergesort, welche die n-elementigen Teillisten zwischen Element $\lfloor \frac{n}{2} \rfloor$ und $\lfloor \frac{n}{2} \rfloor + 1$ auftrennt. Zeichnen Sie einen Rekursionsbaum und annotieren Sie jeden Knoten und jedes Blatt mit dem Ergbnis dieser Auftrennung. Annotieren Sie anschließend jeden inneren Knoten des Rekursionsbaums zusätzlich mit dem Ergebnis der `merge`-Operation.

B.4 Algorithmendesign

Gegeben Sei zusätzlich die Liste $R = \langle 2, 5, 7 \rangle$, welche die geordneten Indizes all derjenigen Elemente in A enthält, welche sich am Ende eines Runs befinden.

B.4.1 Entwurf (2 Punkte)

Implementieren Sie eine Variante von Mergesort `mergesOrt`(A, R), dessen Rekursionstiefe in $\mathcal{O}(|R|)$ liegt. Implementieren Sie den Algorithmus so, dass unter den gegebenen Voraussetzungen nach jedem Rekursionsschritt gilt: "A ist sortiert" g.d.w. $|R| = 1$.

B.4.2 Zusatzfrage (1 Punkt)

Würde es die asymptotische worst-case Laufzeit Ihres Algorithmus verändern, falls die Laufzeit des Initialisierungsschritts, in dem R berechnet wird, mit berücksichtigt wird? Begründen Sie Ihre Antwort.

B.4.3 Ausführen (1 Punkt)

Geben Sie einen Rekursionbaum mit Annotationen wie in Aufgabe B.3 für den Algorithmus in Aufgabe B.4 an.

C Hashing: Nachschlag (1 Punkt)

Ein Mitarbeiter des Instituts für abgewandte Informatik hat eine neue Theorie zum Hashing entwickelt:

Es sei U ein Universum von Eingaben, $\mathcal{H} \subseteq \{0, \ldots m-1\}^U$ eine Familie von Hashfunktionen, und T eine Hashtabelle der Größe m. Die Familie der Hashfunktionen $\mathcal{H}$ sei so gewählt, dass für jedes $u \in U$ und $j \in \{0, \ldots m-1\}$ bei zufälliger gleichverteilter Wahl von $h \in \mathcal{H}$ die Wahrscheinlichkeit $\mathbb{P}[h(u) = j] = \frac{1}{m}$ ist.

Die Behauptung ist nun, dass falls diese Bedingungen erfüllt sind, die erwartete Anzahl von Kollisionen für ein Element u beim Hashen von m Elementen in $\mathcal{O}(1)$ liegt.

Hat dieser Mitarbeiter recht oder liegt er falsch? Liefern Sie zu der Antwort einen Beweis oder ein Gegenbeispiel.

Karlsruher Institut für Technologie

Algorithmen 1 – Sommersemester 2018
Zuverlässige Softwaresysteme im Kontext der Automobilindustrie
`http://verialg.iti.kit.edu`
Prof. Dr. Carsten Sinz – Markus Iser

Deckblatt Übungsblatt 5

Tutoriumsnummer:

Name	Matrikelnummer	Unterschrift

Mit unseren Unterschriften bestätigen wir, dass wir die Aufgaben eigenständig gelöst haben.

Hinweis: Das Übungsblatt darf in Gruppen von bis zu zwei Personen bearbeitet werden. Beide Personen müssen demselben Tutorium zugeteilt sein. Möchte jemand seine Abgaben-Gruppe innerhalb des Semesters wechseln, so ist dies im Voraus mit dem Tutor abzusprechen. **Bitte tragen Sie oben *groß* die Nummer Ihres Tutoriums ein.** Die Lösung des Übungsblattes ist in jedem Fall mit diesem Deckblatt abzugeben.

Bewertung (durch Tutor):