Karlsruher Institut für Technologie

Institut für Theoretische Informatik
T.T.-Prof. Dr. Thomas Bläsius
https://scale.iti.kit.edu

Übungsblatt 05

Algorithmen I - Sommersemester 2024

Abgabe im ILIAS bis 31.05.2024, 18:00 Uhr

Die Abgabe erfolgt alleine oder zu zweit als *eine* PDF-Datei über das Übungsmodul in der Gruppe eures Tutoriums im ILIAS. Beschriftet die Abgabe deutlich mit Matrikelnummer(n) und Name(n). Bei Abgaben zu zweit reicht es, wenn eine Person die Abgabe im ILIAS hochlädt.

- Achtet bei handschriftlichen Abgaben auf Lesbarkeit.
- Achtet darauf, ob Algorithmen in Worten oder Pseudocode beschrieben werden sollen.
- Es werden immer asymptotisch möglichst effiziente Algorithmen erwartet, wenn nicht anders angegeben.
- Wenn Korrektheits- oder Laufzeitanalysen gefordert sind, behandelt diese separat von der Algorithmenbeschreibung.

Gesamtpunkte: 20 (+ 3 Bonus)

Aufgabe 1 - Total eindeutig (5 Punkte)

In dieser Aufgabe darfst du davon ausgehen, mithilfe einer universellen Familie von Hashfunktionen Hashmaps verwenden zu können, die erwartet konstante Zugriffszeiten haben.

In GBI hast du gelernt, dass eine Relation R auf Menge M eine Teilmenge von $M \times M$ ist. Des Weiteren hast du gelernt, dass eine linkstotale[1] und rechtseindeutige[2] Relation auch Funktion genannt wird. Eine andere Eigenschaft von Relationen, die du kennengelernt hast, ist die Symmetrie[3]. Sei nun $|M| = m$ und $|R| = n$.

Beim Forschen an verschiedenen Relationen stellst du fest, dass es für eine gegebene Menge von Tupeln gar nicht so einfach ist festzustellen, ob diese eine der obigen Eigenschaften erfüllt.

a) Entwirf einen Algorithmus, der bei Eingabe Grundmenge M und Tupelmenge R in erwartet $\mathcal{O}(n + m)$ Zeit testet, ob R eine linkstotale Relation auf M darstellt. (1.5 Punkte)

[1] $\forall x \in M : \exists y \in M : (x, y) \in R$
[2] $\forall x, y_1, y_2 \in M : ((x, y_1) \in R \wedge (x, y_2) \in R) \Rightarrow y_1 = y_2$
[3] $\forall x, y \in M : (x, y) \in R \Rightarrow (y, x) \in R$

b) Entwirf einen Algorithmus, der bei Eingabe von R in erwartet $\mathcal{O}(n)$ Zeit testet, ob R eine rechtseindeutige Relation darstellt. Begründe die Laufzeit und Korrektheit deines Algorithmus. (2 Punkte)

c) Entwirf einen Algorithmus, der bei Eingabe von R in erwartet $\mathcal{O}(n)$ Zeit testet, ob R eine symmetrische Relation darstellt. (1.5 Punkte)

Aufgabe 2 - Konstantes Chaos (8 Punkte)

In dieser Aufgabe darfst du davon ausgehen, mithilfe einer universellen Familie von Hashfunktionen Hashmaps verwenden zu können, die erwartet konstante Zugriffszeiten haben. Außerdem darfst du Arrays und Listen verwenden.

In Dr. Metas Geheimlabor herrscht Chaos. Alle Biber watscheln wild durch die Gegend und stören wichtige Experimente. Nachdem er wieder etwas für Ordnung gesorgt hat, möchte Dr. Meta sicherstellen, dass noch alle benötigten Materialien für seine Experimente vorhanden sind. Glücklicherweise hat er alle Sorten von Materialien von 1 bis m durchnummeriert. Das heißt jede Box mit Materialien ist mit der Nummer seiner Sorte versehen. Dafür benötigt er verschiedene Datenstrukturen.

a) Als erstes möchte Dr. Meta wissen, ob bestimmte Sorten noch vorhanden sind. Entwirf daher eine Datenstruktur, die es dir erlaubt Zahlen von 1 bis m in konstanter Zeit einzufügen und zu entfernen, sowie in konstanter Zeit zu prüfen, ob die Zahl mindestens einmal in der Datenstruktur enthalten ist. Hierbei soll die Datenstruktur auch damit umgehen können, dass die selbe Zahl mehrmals eingefügt wird. Erkläre außerdem, wie die drei Operationen umgesetzt werden. (2 Punkte)

Nachdem Dr. Meta deine Datenstruktur genutzt hat, muss er mit Schrecken feststellen, dass nur von n der ursprünglichen m Sorten überhaupt noch Materialien vorhanden sind. Hierbei ist $n \ll m$ und die noch vorhandenen Sorten können beliebig auf die Nummern 1 bis m verteilt sein. Er möchte sich nun einen Überblick darüber verschaffen, wie viele Boxen von jeder Sorte übriggeblieben sind.

b) Beschreibe nun eine Datenstruktur, die es dir erlaubt die Nummern der noch vorhandenen Boxen in erwartet konstanter Zeit einzufügen und zu entfernen, sowie in erwartet konstanter Zeit die Anzahl der in der Datenstruktur verwalteten Boxen einer Sorte abzufragen. Die Datenstruktur soll nur $\mathcal{O}(n)$ Platz verwenden. Beschreibe wie zuvor auch die Umsetzung der drei Operationen. (2 Punkte)

Die Ergebnisse der Untersuchung sind erschreckend. Von jeder Sorte sind nur noch weniger als n Boxen übrig geblieben.

c) Dir steht nun ein Array zur Verfügung, das für jede noch vorhandene Sorte deren Nummer und Anzahl an Boxen als Tupel speichert. Du sollst nun eine Datenstruktur entwerfen, die die Anzahl der Boxen verwaltet. Diese Datenstruktur soll in erwartet $O(n)$ Zeit erstellt werden können und dir erlauben in konstanter Zeit, für

gegebenes k, die k-häufigste Sorte[4] abzufragen. Des Weiteren sollen Änderungen am Datenbestand in erwartet konstanter Zeit möglich sein. Wenn Dr. Meta eine weitere Box in den Untiefen seines Labors findet, soll die Anzahl der entsprechenden Sorte um eins erhöht werden. Wenn die Biber eine Box zerstören, soll die Anzahl der entsprechenden Sorte um eins gesenkt werden. Beschreibe die Datenstruktur und ihre Operationen. Begründe außerdem die zum Erstellen benötigte Zeit und die Laufzeit der Operationen. (4 Punkte)

Hinweis 1: Gehe für diese Aufgabe davon aus, dass zu keinen Zeitpunkt mehr als drei Sorten, die selbe Anzahl an Boxen haben, existieren.

Hinweis 2: In dem Fall, dass mehrere Sorten gleich häufig vorkommen, ist die k-häufigste Sorte nicht mehr eindeutig. In diesem Fall darf innerhalb der gleichhäufig vorkommenden Elemente eine beliebige Reihenfolge gewählt werden. Beispiel: Wenn die Sorten a und b jeweils 4 mal vorkommen und die Sorte c 3 mal, wäre a und b jeweils eine korrekte Antwort für die häufigste und auch zweithäufigste Sorte. Für die dritthäufigste Sorte kommt nur c in Frage.

Aufgabe 3 - Es ist nicht alles eine gute Hashfunktion, was Chaos stiftet (3 Punkte)

Sei $M \in \mathbb{N}_+$, $U = \{0, 1, \ldots, M-1\}$ ein Universum an Schlüsseln und $m \ll M$ die Größe einer Hashtabelle. Gib für jede der folgenden „Hashfunktionen“ an, welche Nachteile bei der Verwendung auftreten können.

a) $h_1 : x \mapsto (4 \cdot x + 3) \mod m$ für gerades m (1 Punkt)

b) $h_2 : x \mapsto ((5 \cdot x + 4) \mod m) + 1$ (1 Punkt)

c) $h_3 : x \mapsto (x + \textsc{dice}(1, 6)) \mod m$
wobei DICE bei jedem Aufruf jeweils mit Wahrscheinlichkeit $1/6$ die Zahlen $1, 2, 3, 4, 5, 6$ ausgibt. (1 Punkt)

Aufgabe 4 - Fast keine Kollisionen bei Familiengeburtstagen (4 Punkte)

Sei $M \in \mathbb{N}_+$, $U = \{0, 1, \ldots, M-1\}$ ein Universum an Schlüsseln und $m \ll M$ die Größe einer Hashtabelle.

a) Sei $\mathcal{H} = \{h : x \mapsto (42x + a) \mod m \mid a \in U\}$.
Zeige, dass $\mathcal{H}$ **keine** universelle Familie ist. (2 Punkte)

b) Sei $\mathcal{H} = \{h : x \mapsto (a \cdot x^3) \mod m \mid a \in U\}$.
Zeige, dass $\mathcal{H}$ **keine** universelle Familie ist. (2 Punkte)

[4] Eine Sorte, die in einer absteigend sortierten Reihenfolge an k-ter Stelle steht, ist eine zulässige k-häufigste Sorte.

Aufgabe 5 - Zum Knobeln (0 Punkte + 3 Bonus)

Wir betrachten nun erneut das Szenario aus Aufgabe 2 c). Wir wollen die Anzahl der Duplikate in der Datenstruktur nicht mehr durch drei beschränken. Das heißt, dass es nun möglich ist, dass beliebig viele Sorten die selbe Anzahl an Boxen haben.

a*) Erkläre kurz, warum eine Datenstruktur aus Aufgabe 2 c) (die den Hinweis nutzt), bei beliebig vielen Duplikaten falsche Ergebnisse bei der Abfrage der k-häufigsten Sorte liefern kann oder die Laufzeit nicht einhält. (1 Punkt)

b*) Beschreibe nun eine Datenstruktur, die die selben Eingaben erhält und die selben Operationen umsetzt, wie in Aufgabe 2 c). Begründe außerdem auch, warum die Laufzeiten der Operationen und des Erstellens weiterhin die obigen Schranken erfüllen. (2 Punkte)

Hinweis: Eine mögliche Lösung ist es die Datenstruktur aus Aufgabe 2 c) zu erweitern, so dass diese mit Duplikaten umgehen kann.

Dr. Meta hat sich bereits an einen geheimen Urlaubsdamm zurückgezogen, um sich von dem beruflichen Stress der letzten Wochen zu erholen.
Er wünscht eine erholsame Pfingst-Pause!