

Zusatzaufgaben 02

Algorithmen I – Sommersemester 2023

Gesamtpunkte: 35

Aufgabe 1 - Datenstrukturen Zuordnen (15 Punkte)

Wir haben im Laufe der Vorlesung mehrere Datenstrukturen kennengelernt. In dieser Aufgabe wollen wir uns auf die folgenden fünf konzentrieren:

- Arrays
- (doppelt-verkettete) Listen
- Stacks
- Queues
- Hashtabellen

Deine Aufgabe ist es nun, jedem der folgenden Szenarien eine dieser Datenstrukturen zuzuordnen, die sich aus deiner Sicht besonders dafür eignet. Begründe dabei kurz deine Wahl.

1. Für ein Programm soll eine Undo/Redo-Funktion entwickelt werden, bei dem die vergangenen Veränderungen gespeichert werden sollen.
2. Für einen Online-Shop soll für ein Produkt (welches durch seine Produktnummer eindeutig identifizierbar ist) vor dem Verkauf überprüft werden, wie viele Exemplare noch vorhanden sind.
3. In einer App gibt es einen Content-Feed, bei dem man nach rechts/ links swipen kann, um aktuellere/ältere Nachrichten zu erhalten.
4. Für eine Veranstaltung mit 5000 durchnummerierten Tickets soll am Einlass kontrolliert werden, welche Tickets bereits entwertet wurden.

5. Für eine Anwendung soll jeweils die letzte noch nicht gelesene Nachricht auf eurem Telefon angezeigt werden.
6. Die Bücher auf einem TBR-Stapel¹ sollen organisiert werden. Dabei ist immer nur das Buch relevant, das ganz oben auf dem Stapel liegt.
7. Während der Sprechstunde sollen die Patienten in einem Wartezimmer verwaltet werden. Die Patienten werden dabei in der Reihenfolge aufgerufen, wie sie in die Praxis gekommen sind.
8. Für eine Anwendung sollen Bewegungen zwischen Zugabteilen bei einem Zug beliebiger Länge simuliert werden. Es soll dabei gespeichert werden, wie viele Personen sich zu einem Zeitpunkt in einem Zugabteil befinden.
9. In einer Bar sind leider schon einige Menschen auffällig geworden und haben Hausverbot erhalten. Um dies festzuhalten wurden ihre den Vor- und Nachnamen und die Ausweisnummer des Personalausweises gespeichert. Für eine Person soll nun festgestellt werden können, ob sie schon einmal Hausverbot erhalten hat.
10. Für jeden Knoten in einem Graphen soll eine Farbe gespeichert werden.
11. Bei einem rundenbasierten Online-Game ohne feste Spieleranzahl soll die Spieler-Reihenfolge gespeichert werden.
12. Für eine neue Speedrun-Website soll es pro Spiel ein Leaderbord geben, in dem die 10 besten Spieler stehen sollen.
13. Für einen Music-Player soll eine Playlist abgespielt werden. Dabei soll es auch möglich sein, einen vergangenen Song anzuhören oder Songs zu skippen.
14. Zur Ausgabe eines Browserverlaufs sollen die zuletzt besuchten Websites gespeichert werden.
15. Für eine passwortgeschützte Website sollen für Neukunden Profile angelegt werden. Bei späterem Einloggen soll überprüft werden, ob ein eingegebenes Passwort richtig war. Um Hacking zu vermeiden sollte das Passwort allerdings nicht gespeichert werden.

¹TBR steht für „to be read“, hier liegen also alle Bücher, die man schon seit Monaten ganz bestimmt sehr bald lesen wird.

16. In einem digitalen Kochbuch sollen die Schritte eines Rezepts gespeichert werden. Dabei wird dem Anwender immer nur der aktuelle Schritt angezeigt.

Aufgabe 2 - Immer in Paaren laufen (7 Punkte)

Im Folgenden wollen wir in erwarteter Linearzeit Paare von ganzen Zahlen in einem Array finden, die eine gewisse Eigenschaft erfüllen.

1. Beschreibe einen Algorithmus, der für eine gegebene Zahl k und ein Array A mit n Einträgen in erwarteter $\mathcal{O}(n)$ Zeit entscheidet, ob es Indizes i, j gibt, sodass $A[i] \cdot A[j] = k$. (2 Punkte)
2. Beschreibe einen Algorithmus, der für eine gegebene Zahl k und ein Array A mit n Einträgen in erwarteter $\mathcal{O}(n)$ Zeit entscheidet, ob es Indizes i, j gibt, sodass $\frac{2 \cdot A[i]}{3} + 4 \cdot A[j] + 5 = k$. (2 Punkte)
- *. Finde ein generelles Verfahren, womit wir für eine gegebene Zahl $k \in \mathbb{Z}$, ein Array A mit n Einträgen und eine Funktion $f(x, y) : \mathbb{Z} \times \mathbb{Z} \rightarrow \mathbb{Z}$ in erwarteter $\mathcal{O}(n)$ entscheiden können, ob es zwei Indizes i, j gibt, sodass $f(A[i], A[j]) = k$. Welche Anforderungen muss die Funktion f erfüllen? (3 Punkte)

Aufgabe 3 - Familientreffen (4 Punkte)

In dieser Aufgabe wollen wir möglicherweise universelle Hashfamilien untersuchen. Sei dafür $M \in \mathbb{N}$ und $U = \{0, 1, \dots, M - 1\}$ unser Universum an Schlüsseln und $m \ll M$ die Größe unserer Hashtabelle.

1. Ist die folgende Menge an Funktionen eine universelle Familie an Hashfunktionen? (2 Punkte)

$$\mathcal{F} = \{f : x \mapsto a^x \bmod a \bmod m \mid a \in U\}$$

2. Ist die folgende Menge an Funktionen eine universelle Familie an Hashfunktionen? (2 Punkte)

$$\mathcal{H} = \{h : x \mapsto (ax + 5) \bmod m \mid a \in U\}$$

Aufgabe 4 - Auf heißer Spur (9 Punkte)

Alle Tiere sind vereinigt im Aufstand gegen den hinterhältigen Dr. Meta. Alle Tiere? Nein! Eine Hand voll Dachse steht immer noch an der Seite des Superschurken (vermutlich diejenigen, die hoffen, irgendwann von ihm noch ihre Smartphones zurück zu bekommen). Natürlich hat Dr. Meta Wind von der Intrige in den Alpen bekommen. Außer sich vor Wut schickt er sämtliche ihm verbleibenden Lakaien auf die Baustelle, um die Fertigstellung des Damms zu sabotieren.

Dr. Meta hat jedoch die Rechnung ohne die Seeadler gemacht. Sie fliegen unablässig Patrouille und notieren sich verdächtige Dachse. Sie haben nur ein Problem: Selbst die auf höchstem Niveau ausgebildeten Agenten des NSA können Dachse nicht voneinander unterscheiden. Otter, Dachs, Biber... die sehen doch alle gleich aus mit ihrem Fell! In den Augen der Adler sind sie alle Beute Säugetiere. Um herauszufinden, welchen Dachs er gesichtet hat, muss ein Seeadler also den Verdächtigten befragen.

Das nutzen die von Dr. Meta eingeschleusten Dachse geschickt aus: Werden sie von verschiedenen Seeadlern nach ihrem Namen gefragt, können sie verschiedene Antworten geben. Zwei Seeadler können den gleichen Dachs also unter verschiedenen Namen kennen. Um nicht zu sehr aufzufallen, benutzen verschiedene Dachse aber nie den gleichen Decknamen.

Doch der NSA wäre nicht der NSA, könnte er damit nicht umgehen. Die Störenfriede müssen schnellstmöglich identifiziert und von der Baustelle entfernt werden!

Aktuell liegen dem NSA sämtliche Sichtungen von verdächtigen Dachsen in Form einer Liste `sightings` der Länge n vor. Ein Eintrag (t, c, d) in `sightings` drückt aus, dass zum Zeitpunkt t an Koordinate c ein Dachs gesichtet wurde, der behauptet hat, den Decknamen d zu tragen. Damit wissen die Adler: Gibt es zwei Einträge $(t, c, d_1), (t, c, d_2)$ in `sightings`, sodass $d_1 \neq d_2$, so war der gesichtete Dachs ein Spion, der zwei Adler-Agenten unterschiedliche Namen genannt hat.

Zunächst will der NSA wissen, welche Decknamen überhaupt in Umlauf sind.

1. Beschreibe, wie eine Liste sämtlicher Decknamen in `sightings` erstellt werden kann, die keine Duplikate enthält. Dein Algorithmus soll dabei eine asymptotische Laufzeit haben, die besser als $\Theta(n^2)$ ist. (2 Punkte)

Das ist doch gleich viel übersichtlicher! Die Adler sind jedoch nicht unfehlbar (auch wenn sie das nie zugeben würden) und befragen auch unschuldige Dachse, deren Namen damit auch in `sightings` auftauchen. Welche Decknamen gehören nun zu welchem Spitzel? Und wie viele Maulwürfe sind tatsächlich unter den Dachsen?

2. Wir wollen im Folgenden einen Algorithmus entwerfen, der die Namen in `sightings` zu Dachsen zuordnet. Welche Datenstruktur eignet sich am Besten, um die Namen dazu zu verwalten? Begründe deine Wahl. (1 Punkt)
3. Beschreibe einen Algorithmus, der für alle gesichteten Dachse die Menge der von ihnen angegebenen Namen bestimmt. (3 Punkte)
4. Gib die asymptotische Laufzeit deines Algorithmus aus Teilaufgabe 3 an und begründe deine Antwort. (2 Punkte)
5. Beschreibe einen Algorithmus, der die Anzahl an Dachsen bestimmt, die auf jeden Fall zu Dr. Metas Spionen gehören. Die Laufzeit dieses Algorithmus sollte nicht schlechter sein als die deines Algorithmus aus Teilaufgabe 3. (1 Punkt)