

Zusatzaufgaben 02

Algorithmen I – Sommersemester 2022

Gesamtpunkte: 40

Aufgabe 1 - Des Graphen neue Kleider (10 Punkte)

Um einen gerichteten Graphen $G = (V, E)$ in Computerprogrammen darzustellen gibt es verschiedene Möglichkeiten. Die wohl simpelste Darstellung ist dabei die Kantenliste E . Aus der Kantenliste kann sehr einfach V berechnet werden:

$$V = \bigcup_{(u,v) \in E} \{u, v\}$$

Andere Operationen, wie die Nachbarschaft von $N(v) = \{u \mid (v, u) \in E\}$ von $v \in V$ oder den Grad $\deg(v) = |N(v)|$ zu berechnen, sind in dieser Darstellung jedoch nicht effizient und somit nur in $\mathcal{O}(|E|)$ umsetzbar.

Deswegen werden Graphen häufig in einem *Adjazenzarray* dargestellt. Hierfür nehmen wir an, dass wir die Knoten durchnummerieren können, also $V = \{0, \dots, n-1\}$ gilt. Der Index von $v \in \{0, \dots, n\}$ ist definiert durch:

$$I(v) = \begin{cases} 0 & | \ v = 0 \\ I(v-1) + |N(v-1)| & | \text{sonst} \end{cases}$$

Eine Adjazenzarray-Datenstruktur besteht aus zwei Arrays $\mathcal{V}, \mathcal{E}$. $\mathcal{V}$ speichert die Indizes $\mathcal{V} = \{I(0), I(1), \dots, I(n)\}$. $\mathcal{E}$ speichert alle Nachbarn aller Knoten: $\mathcal{E} = N(v_0) \circ N(v_1) \circ \dots \circ N(v_{n-1})$. Hierbei ist $\circ$ die Verkettung von Arrays (Beispiel: $\{1, 2\} \circ \{3, 4\} = \{1, 2, 3, 4\}$).

1. Gegeben sei $\mathcal{V} = \{0, 2, 4, 7, 8, 8, 10\}$ und $\mathcal{E} = \{1, 2, 0, 4, 1, 3, 5, 1, 2, 4\}$. Zeichne den zugehörigen Graphen. (1 Punkt)

2. Bei genauer Beobachtung fällt auf, dass $|\mathcal{V}| = |V| + 1$. In der Vorlesung haben wir Dummy- bzw. Sentinel-Elemente kennengelernt. Welchen Zweck erfüllen diese? Bei welchen Datenstrukturen aus der Vorlesung haben wir Sentinel-Elemente kennengelernt? Welchen Zweck erfüllt das Sentinel-Element im Adjazenzarray? (2 Punkte)
3. Wie kann in einem Adjazenzarray in $\mathcal{O}(1)$ der Grad eines Knotens v bestimmt werden? Wie können wir in $\mathcal{O}(\deg(v))$ über $N(v)$ iterieren? (2 Punkte)
4. Gib einen Algorithmus in Pseudocode an, der ein Adjazenzarray in $\mathcal{O}(|V| + |E|)$ in eine Kantenliste konvertiert. Begründe, warum dein Algorithmus die Laufzeitanforderung erfüllt. Gegeben sei dazu die folgende Signatur:

$\text{ARRAYTOLIST}(\mathcal{V}: [\mathbb{N}; n + 1], \mathcal{E}: [\mathbb{N}; m]): \text{List}(\mathbb{N} \times \mathbb{N})$

(2 Punkte)

5. Gib einen Algorithmus in Pseudocode an, der eine Kantenliste in ein Adjazenzarray in $\mathcal{O}(|V| + |E|)$ konvertiert. Begründe, warum dein Algorithmus die Laufzeitanforderung erfüllt. Gegeben sei dazu die folgende Signatur:

- n : Anzahl der Knoten, $V = \{1, \dots, n - 1\}$
- m : Anzahl der Kanten
- L : Liste der Kanten

$\text{LISTTOARRAY}(n: \mathbb{N}, m: \mathbb{N}, L: \text{List}(\mathbb{N} \times \mathbb{N})): ([\mathbb{N}; n + 1], [\mathbb{N}; m])$

(3 Punkte)

Aufgabe 2 - Datenstrukturen Zuordnen (15 Punkte)

Gegeben sind insgesamt 15 Szenarien und fünf Datenstrukturen (Array, Listen: Stack, FIFO, (allgemeine) doppelt-verkettete Listen und Hashtabellen). Deine Aufgabe ist es nun, die Szenarien der passenden Datenstruktur zuzuordnen. Dabei gibt es pro Datenstruktur genau drei zugehörige Szenarien. Begründe kurz für jede Datenstruktur, warum Du genau diese Szenarien ausgewählt hast.

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 (gegeben durch Produkt-nummer) vor dem Verkauf überprüft werden, wie viele 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. Es werden (unpriorisiert) E-Mails an einen Newsletter versendet.
7. 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.
8. 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.
9. Für ein Multiple-Choice Quiz mit 20 Fragen je 4 Antwortmöglichkeiten soll ausgegeben werden, wie viele Studenten die entsprechende Antwort ausgewählt haben.
10. Bei einem rundenbasierten Online-Game ohne feste Spieleranzahl soll die Spieler-Reihenfolge gespeichert werden.
11. Für eine neue Speedrun-Website soll es pro Spiel ein Leaderbord geben, in dem die 10 besten Spieler stehen sollen.
12. Für einen Music-Player soll eine Playlist abgespielt werden. Dabei soll es auch möglich sein, einen vergangenen Song anzuhören oder Songs zu skippen.

13. Zur Ausgabe eines Browserverlaufs sollen die zuletzt besuchten Websites gespeichert werden.
14. 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.
15. Ein Nutzer lädt Videos von einer Website herunter. Falls ein Video noch nicht vollständig heruntergeladen ist, soll erst sein Download abgeschlossen werden, bevor ein Video heruntergeladen wird, das erst danach ausgewählt wurde.

Aufgabe 3 - 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 4 - Auf heißer Spur (8 Punkte)

Der brillante und wachsame Superbösewicht Dr. Meta dachte, mit seinen Säuberungsaktionen sämtliche Maulwürfe unter seinen Mitarbeitern entfernt zu haben. Doch weit gefehlt!

Er hat herausbekommen, dass nicht nur der Eindringling, der in sein Labor

eingebrochen ist, sondern auch dessen Komplizen von seiner Erzfeindin Theresa Trivial angeheuert wurden. Kurzerhand macht sich Dr. Meta auf, herauszufinden, was Theresa im Schilde führt. Mit seinen verlässlichsten Spitzeln zusammen infiltriert er die Burg von Theresa Trivial, in der es vor Spionen wimmelt.

Die Spitzel von Dr. Meta positionieren sich in der Burg und notieren jedes Mal, wenn sie einen verdächtigen Spion sehen. Somit entsteht eine Liste mit Tripeln der Form (Zeit, Ort, Name). Doch die Spione sind äußerst gewieft: Werden sie nach ihrem Namen gefragt, geben sie jedem Spitzel von Dr. Meta eine andere Antwort. Somit kennen zwei Spitzel den gleichen Spion unter unterschiedlichen Namen.

Dr. Meta will nun herausfinden, welche Decknamen zu welchem Spion gehören. Gegeben hat er dazu eine Liste L der Länge n , die Tripel der Form (Zeit, Ort, Name) enthält. Ein Tripel (t, p, d) beschreibt dabei, dass der Spion mit Decknamen d zum Zeitpunkt t am Ort p gesichtet wurde. Gibt es zudem einen weiteren Eintrag (t, p, d') in L , so weiß Dr. Meta, dass d und d' Decknamen des gleichen Spions sind.

Wir wollen nun einen Algorithmus entwickeln, der die Decknamen in L den Spionen zuordnet.

1. Beschreibe, wie eine Liste sämtlicher Decknamen in L erstellt werden kann, die keine Duplikate enthält. Dein Algorithmus soll dabei eine asymptotische Laufzeit haben, die besser als $\Theta(n^2)$ ist. (2 Punkte)
2. Welche Datenstruktur eignet sich am Besten, um die Decknamen in L zu verwalten? Begründe deine Wahl. (1 Punkt)
3. Beschreibe einen Algorithmus, der für alle gesichteten Spione die Menge ihrer Decknamen bestimmt. (3 Punkte)
4. Gib die asymptotische Laufzeit deines Algorithmus aus Teilaufgabe 3 an und begründe deine Antwort. (2 Punkte)