Karlsruher Institut für Technologie

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

Übungsblatt 14

Algorithmen I – Sommersemester 2022

Abgabe im ILIAS bis 03.08.2022, 14:00 Uhr

Bitte beschrifte Deine Abgabe gut sichtbar mit Deinem Namen und Deiner Matrikelnummer. Achte insbesondere bei handschriftlichen Abgaben auf Lesbarkeit und genügend Platz für Korrektur-Anmerkungen. Die Abgabe erfolgt über das Übungsmodul in der Gruppe Deines Tutoriums im ILIAS. Gib Deine Ausarbeitungen in *einer* PDF-Datei ab. Achte darauf, effiziente Algorithmen zu formulieren, also solche mit möglichst geringer asymptotischer Laufzeit!
Wenn du die Korrektheit eines Algorithmus begründen oder dessen Laufzeit analysieren sollst, tue dies getrennt von der Beschreibung des Algorithmus.
Wenn nicht anders spezifiziert oder aus dem Kontext ersichtlich, bezeichnen wir mit Graph einen einfachen ungerichteten Graphen.

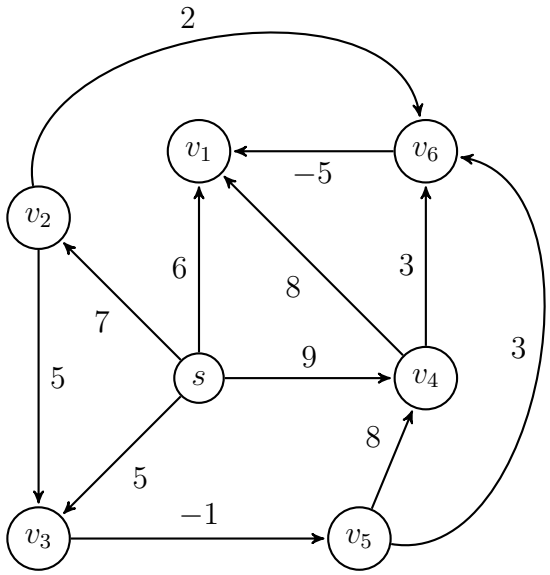
Aufgabe 1 - Baobabs (0+5 Punkte)

Ein *Binärbaum* ist ein Baum, bei dem jeder innere Knoten genau zwei Kinder hat. Gegeben sei ein beliebiger Baum $T = (V, E)$. Beschreibe einen Algorithmus, der einen Teilbaum $T' = (V', E')$ von T ausgibt, sodass T' ein Binärbaum und $|V'|$ maximal ist. Begründe die Korrektheit deines Algorithmus. Nenne und begründe außerdem seine asymptotische Laufzeit. (5 Punkte)

Aufgabe 2 - Sind wir schon da? (0+4 Punkte)

Sei $G = (V, E)$ ein DAG und $s \in V$ die einzige Quelle. Wir wollen nun mittels Bellman-Ford Algorithmus die Distanzen von s zu allen anderen Knoten bestimmen.

1*. Gib für folgenden DAG an, in welcher Reihenfolge die Kanten relaxiert werden müssten, wenn man alle gesuchten Distanzen in der ersten Iteration des Algorithmus bestimmt haben möchte. (1 Punkt)



2*. Gib an und begründe, ob deine in Teilaufgabe 1 angegebene Reihenfolge, die einzig mögliche Antwort ist. (1 Punkt)

3*. Gib für einen beliebigen DAG G an, in welcher Reihenfolge die Kanten relaxiert werden müssten, wenn man alle gesuchten Distanzen in der ersten Iteration des Algorithmus bestimmt haben möchte. (2 Punkte)

Aufgabe 3 - Let's play a game! (0+7 Punkte)

Wir wollen eine Partie Patience mit n Karten spielen. Der Wert einer Karte ist durch eine natürliche Zahl definiert. Ein Spiel läuft wie folgt ab:
Wir beginnen mit einem unsortierten Deck von n Karten. Diese sollen in sortierte Stapel eingeordnet werden, welche von links nach rechts angeordnet auf dem Tisch liegen. Wir nehmen die Karten nacheinander vom Deck und führen eine der folgenden Aktionen aus: Sei dabei w der Wert der aktuellen Karte.

- Wurden noch keine Stapel angelegt oder die jeweils oberste Karte jedes Stapels hat einen Wert kleiner w, wird rechts von den bereits bestehenden Stapeln ein neuer Stapel angelegt und die aktuelle Karte dort eingefügt.
- Gibt es mindestens einen Stapel, dessen oberste Karte einen Wert von mindestens w hat, legen wir die aktuelle Karte auf den linkesten dieser Stapel.

1*. Das folgende Array stellt ein Deck dar, wobei jeder Eintrag eine Karte repräsentiert. Gib die Stapel an, die sich durch das Einsortieren der Karten wie oben beschrieben ergeben. (1 Punkt)

$$\langle 5, 4, 1, 8, 8, 7, 3, 4, 3, 7, 5 \rangle$$

2*. Beschreibe einen Algorithmus, der die Karten, nachdem sie wie beschrieben in Stapel eingeordnet wurden, zu einem einzigen sortierten Stapel zusammengefügt. (1 Punkt)

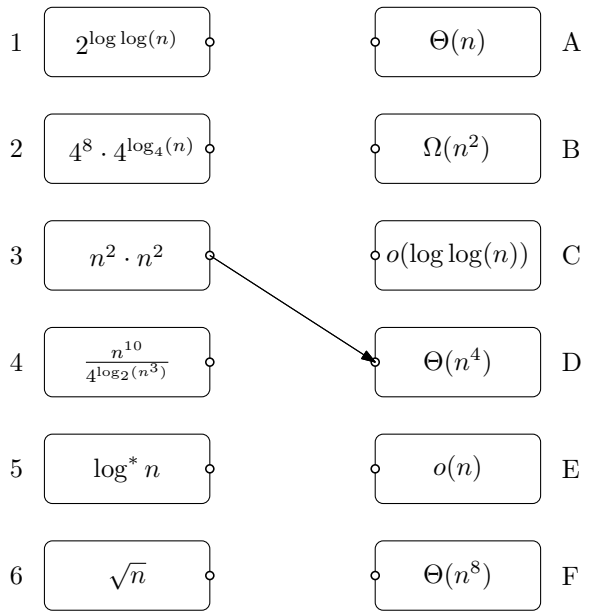
3*. Gegeben sei ein Array `cards` : $[\mathbb{N};\ n]$, welches eine unsortierte Menge Spielkarten repräsentiert. Zudem ist eine Methode RECONSTRUCT gegeben, die deinen Algorithmus aus Teilaufgabe 2 (Stapel zusammenführen) umsetzt. Gib einen Algorithmus in Pseudocode an, der als Eingabe `cards` erhält, Patience spielt und schließlich das die sortierte Kartenmenge zurückgibt. Verwende die folgende Signatur: (3 Punkte)

PATIENCE(`cards`: $[\mathbb{N};\ n]$): $[\mathbb{N};\ n]$

4*. Nenne und begründe die Laufzeit deines Algorithmus und ob er stabil ist. (2 Punkte)

Aufgabe 4 - "Oh no"-Tation (0+4 Punkte)

Im folgenden Graphen soll es von jedem Knoten auf der linken Seite $(1, 2, 3, 4, 5, 6)$ genau dann eine gerichtete Kante zu einem Knoten auf der rechten Seite (A, B, C, D, E, F) geben, wenn die zugehörige Funktion in der jeweiligen Menge enthalten ist. So beschreibt die eingezeichnete Kante (3, D), dass $n^2 \cdot n^2 \in \Theta(n^4)$ gilt. Zeichne die fehlenden Kanten ein, oder gib die zugehörigen Tupel aus Zahlen und Buchstaben an.
Hinweis: Jede fehlende und jede falsche Kante geben einen halben Punkt Abzug. (4 Punkte)



Aufgabe 5 - Daten, Daten (0+5 Punkte)

Bestimme für folgende Situationen, welche möglichst einfache Datenstruktur geeignet ist und begründe deine Wahl kurz.

Hinweis: In der Vorlesung haben wir folgende Datenstrukturen kennengelernt: Listen, dynamische Arrays, Hashtabellen, binäre Heaps, $(2, 3)$-Bäume, Union-Find, Adjazenzliste.

1*. Den durchnummerierten Knoten in einem Graphen sollen Farben zugeordnet werden, welche anschließend anhand der Knoten-ID ausgelesen werden können. (1 Punkt)

2*. Bei einer Volkszählung geben Leute ihren Wohnort an. Dabei soll zu jedem Zeitpunkt bestimmt werden können, wie viele Leute aus einem gegebenen Ort sich bereits gemeldet haben. (1 Punkt)

3*. Ein Pizza-Lieferant bekommt im Laufe des Abends Bestellungen, die in der Reihenfolge abgearbeitet werden, in der sie ankommen. (1 Punkt)

4*. Für jede Stadt soll gespeichert werden, welche Krankenhäuser sich im Umkreis von 25km befinden. (1 Punkt)

5*. Bei einer Verlosung haben sich viele Glückspilze registriert. Jede Woche wird ein $k \in \mathbb{N}$ zufällig bestimmt und Preise an jede k-te registrierte Person verschenkt, wobei die Reihenfolge betrachtet wird, in der sich die Leute registriert haben. (1 Punkt)