Lehrstuhl für Informatik I
Algorithmen und Komplexität
Universität Würzburg

Würzburg, den 20. Januar 2022

Prof. Dr. Alexander Wolff
Felix Klesen, M. Sc.

9. Übungsblatt zur Vorlesung Algorithmen und Datenstrukturen (Winter 2021/22)

Aufgabe 1 – Zweifärbbarkeit

Ein Graph $G = (V, E)$ heißt *zweifärbbar*, wenn eine Abbildung $c\colon V \to \{\text{rot}, \text{blau}\}$ existiert, so dass für jede Kante $\{u, v\} \in E$ gilt, dass $c(u) \neq c(v)$. Zwei zueinander benachbarte Knoten erhalten also stets unterschiedliche Farben.

a) Welches ist der kleinste Graph, der nicht zweifärbbar ist? **1 Punkt**

b) Entwerfen Sie einen Algorithmus in Pseudocode, der für einen gegebenen Graphen $G = (V, E)$ und eine gegebene Färbung c testet, ob es zwei benachbarte Knoten mit derselben Farbe gibt. Was ist die asymptotische Worst-Case-Laufzeit des Algorithmus? **2 Punkte**

c) Entwerfen Sie einen Algorithmus in Pseudocode, der für einen gegebenen Graphen $G = (V, E)$ ermittelt, ob er zweifärbbar ist. Die Laufzeit des Algorithmus soll $O(|V| + |E|)$ sein. Achtung: G ist nicht notwendigerweise zusammenhängend. **3 Punkte**

Aufgabe 2 – Tiefensuche

Geben Sie für jedes der geforderten Beispiele den Graphen und das Ergebnis einer Tiefensuche in Form der resultierenden Bäume sowie der Entdeckungs- und Abschlusszeiten an. In dieser Aufgabe sind keine Selbst- oder Mehrfachkanten erlaubt.

a) Geben Sie ein Beispiel an, das folgende Behauptung widerlegt:

Sei G ein gerichteter Graph, der einen Pfad von u nach v enthält, und sei $u.d < v.d$ das Resultat einer Tiefensuche in G. Dann folgt, dass v im Tiefensuchbaum ein Nachkomme von u ist (d.h. es gibt in diesem Baum einen u-v-Pfad). **2 Punkte**

b) Geben Sie ein Beispiel an, das folgende Behauptung widerlegt:

Sei G ein gerichteter Graph, der einen Pfad von u nach v enthält. Für die Entdeckungs- und Abschlusszeiten jeder Tiefensuche in G gilt dann $v.d < u.f$. **1 Punkt**

c) Geben Sie ein Beispiel für eine Tiefensuche in einem gerichteten Graphen G an, in der ein Baum mit einem einzelnen Knoten u gebildet wird, obwohl u sowohl eingehende als auch ausgehende Kanten hat. **2 Punkte**

Aufgabe 3 – Schlangen

Gegeben sei eine Schlange entsprechend der Definition in der Vorlesung.

a) Erweitern Sie die Datenstruktur um eine Methode *int* Rank(*key* k), die bestimmt, nach wie vielen Aufrufen der Methode Dequeue das gesuchte Element ausgegeben würde, ohne die Schlange dabei zu verändern. Schreiben Sie kommentierten Pseudocode mit Laufzeit $O(n)$. **4 Punkte**

b) Wieso ist es nicht möglich die Schlange mit $o(n)$ zusätzlichem Speicher so zu augmentieren, dass Rank-Anfragen in $o(n)$ Zeit beantwortet werden können? **1 Punkt**

c) Nehmen Sie jetzt an, dass wir die Schlange mit einem zusätzlichen binären Suchbaum augmentieren um die Anfragen in $O(\log n)$ beantworten zu können. Welche Auswirkungen hat dessen Aufrechterhaltung auf die Laufzeiten der ursprünglichen Schlangenoperationen? **1 Punkt**

d) Augmentieren Sie die ursprüngliche Schlange mit $O(n)$ zusätzlichem Speicher so, dass die Anfragen in $O(1)$ beantwortet werden können, ohne die Laufzeit anderer Methoden zu beeinflussen. Sie dürfen dafür annehmen, dass eine injektive Abbildung von *key* nach $\{1, 2, \ldots, O(n)\}$ existiert. **3 Punkte**

Bitte geben Sie Ihre Lösungen bis **Donnerstag, 27. Januar 2022, 14:00 Uhr** einmal pro Gruppe über Wuecampus als pdf-Datei ab. Vermerken Sie dabei stets die Namen und Übungsgruppen aller BearbeiterInnen auf der Abgabe.

Grundsätzlich sind stets alle Ihrer Aussagen zu begründen und Ihr Pseudocode ist stets zu kommentieren.

Die Lösungen zu den mit PABS gekennzeichneten Aufgaben, geben Sie bitte nur über das PABS-System ab. Vermerken Sie auf Ihrem Übungsblatt, in welchem Repository (*sXXXXXX*-Nummer) die Abgabe zu finden ist. Geben Sie Ihre Namen hier als Kommentare in den Quelltextdateien an.