

8. Übungsblatt zur Vorlesung Algorithmen und Datenstrukturen (Winter 2019/20)

Aufgabe 1 – Schlange aus zwei Stapeln analysieren

Zwei Stapel S_1, S_2 können wie folgt eine Schlange Q simulieren:

- **Enqueue(x):** führe $S_1.\text{Push}(x)$ aus – lege also x auf den ersten Stapel.
- **Dequeue():** Ist S_2 leer, führe solange $S_2.\text{Push}(S_1.\text{Pop}())$ aus, bis S_1 leer ist – verschiebe die Elemente der Reihe nach von S_1 auf S_2 . Danach führe $S_2.\text{Pop}()$ aus.
- **Empty():** gibt true zurück, falls S_1 und S_2 leer sind, sonst false .

Betrachten Sie nun eine zufällige Folge der Länge n bestehend aus **Enqueue**-, **Dequeue**- und **Empty**-Operationen auf Q .

- a) Argumentieren Sie, warum die *Worst-Case*-Laufzeit einer einzelnen **Dequeue**-Operation auf Q in $\Theta(n)$ liegt. **3 Punkte**
- b) Zeigen Sie mit amortisierter Analyse: die Gesamtlaufzeit für jede Folge liegt ebenfalls in $\Theta(n)$. Eine einzelne **Dequeue**-Operation benötigt amortisiert also nur konstante Laufzeit. Wieso ist dies kein Widerspruch zu Teilaufgabe a)? **3 Punkte**

Aufgabe 2 – Zweifärbbarkeit

Ein Graph $G = (V, E)$ heißt *zweifärbbar*, wenn eine Abbildung $c: V \rightarrow \{\text{rot}, \text{blau}\}$ 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 3 – Pfade in kreisfreien Graphen

Gegeben sei ein gerichteter kreisfreier Graph $G = (V, E)$, zwei Knoten $s, t \in V$ und eine Menge $W \subset V$ von k Knoten, wobei $s, t \notin W$. Wir bezeichnen einen s - t -Pfad P als *zulässig*, wenn er durch alle Knoten in W geht. Dabei ist es egal, in welcher Reihenfolge die Knoten in W durchlaufen werden.

- a) Seien P_1 und P_2 zwei zulässige s - t -Pfade. Zeigen Sie, dass P_1 und P_2 die Knoten in W in derselben Reihenfolge durchlaufen. **1 Punkt**
- b) Ergänzen Sie den folgenden Algorithmus, so dass er einen zulässigen s - t -Pfad als Liste von Knoten zurückgibt, falls es solch einen Pfad gibt. Falls es keinen zulässigen s - t -Pfad gibt, soll der Algorithmus `nil` zurückgeben. Begründen Sie, warum der Algorithmus korrekt ist. **3 Punkte**

```
getPath(DirectedAcyclicGraph G, Vertex s, Vertex t, Array of Vertices W)
```

```
  P = new List()
```

```
  A = getVertexOrder(G, s, t, W) // Liefert ein Feld mit s, mit den Knoten in W in  
    der richtigen Reihenfolge, wenn es eine solche gibt, und mit t.
```

```
  for i = A.length downto 2 do
```

```
    // Ergänze Pfad P um Knoten zwischen A[i - 1] und A[i]  
    // (ohne A[i - 1] aber einschließlich A[i]).  
    // Gib nil zurück, wenn es von A[i - 1] nach A[i] keinen Pfad gibt.
```

```
  P.Insert(A[1])
```

```
  return P
```

- c) Hat Ihr Algorithmus aus Aufgabenteil b) die Laufzeit $O(|V| + |E|)$? Falls nicht, beschreiben Sie in Worten, wie man diese Laufzeit durch Verbesserung Ihres Algorithmus erreichen kann. **2 Punkte**
- d) Gegeben sei ein gerichteter Graph $G = (V, E)$, der nicht notwendigerweise kreisfrei ist, zwei Knoten $s, t \in V$ und eine Folge von Knoten $W = (w_1, w_2, \dots, w_n)$. Zeigen Sie, dass im schlechtesten Fall jeder s - t -Pfad, der die Knoten in W durchläuft und dabei die gegebene Reihenfolge einhält, Länge $\Omega(|V|^2)$ hat. Beachten Sie, dass ein Pfad einen Knoten mehrfach durchlaufen kann. **2 Punkte**

Bitte werfen Sie Ihre Lösungen bis **Donnerstag, 9. Januar 2020, 13:00 Uhr** in den Vorlesungs-Briefkasten im Informatik-Gebäude. Geben Sie stets die Namen und Übungsgruppen aller BearbeiterInnen sowie die Übungsgruppe, in der das Blatt zurückgegeben werden soll, an.

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.