

## Algorithmen und Programmierung 3, WS 2019/2020 — 2. Übungsblatt

Abgabe bis Freitag, 1. November 2019, 12:00 Uhr, in die Fächer der Tutoren

---

9. Topologisches Sortieren, 0 Punkte

Geben Sie alle möglichen topologischen Sortierungen für folgende Eingabe an: Es gibt 6 Knoten, und die Kanten sind  $\{(6, 4), (3, 5), (4, 2), (3, 4), (5, 4), (1, 2), (1, 6), (3, 6)\}$ .

10. Topologisches Sortieren, 0 Punkte

- (a) Was passiert beim in der Vorlesung angegebenen Algorithmus für topologisches Sortieren, wenn ein Paar  $(i, j)$  in der Eingabe mehrfach auftritt? Was passiert, wenn ein Paar  $(i, i)$  auftritt?
- (b) Untersuchen Sie verschiedene Möglichkeiten, wie man die Liste der freien Elemente beim topologischen Sortieren verwalten kann, im Hinblick auf ihre Effizienz. Kann man auf diese Liste auch gänzlich verzichten?
- (c) Welche Variablen oder Attribute im Programm aus der Vorlesung<sup>1</sup> könnte man ohne Nachteil einsparen?

11. Zirkuläre Abhängigkeit, 7 Punkte

Erweitern Sie den Algorithmus zum topologischen Sortieren aus der Vorlesung, sodass bei der Existenz eines gerichteten Kreises nicht einfach mit einer Meldung abgebrochen wird, sondern auch ein Kreis (als „Beweis“) ausgegeben wird.

Beschreiben Sie Ihren erweiterten Algorithmus in Pseudo-Code. (Zum Beispiel können Sie eine Schleife „für alle Elemente  $x \dots$ “ schreiben.)

Erklären Sie Ihren Algorithmus *in Worten*, ohne notwendigerweise auf Details der Implementierung einzugehen. (Sie müssen nur das erklären, was gegenüber dem ursprünglichen Algorithmus neu ist. Ein Korrektheitsbeweis ist nicht verlangt.)

12. Programmieraufgabe, 6 Punkte, hochzuladen auf der KVV-Seite der Veranstaltung

Schreiben Sie ein Java-Programm für die vorige Aufgabe. Sie können das Programm aus der Vorlesung<sup>1</sup> erweitern. Ihr Programm soll nicht mehr als  $O(m+n)$  zusätzliche Zeit und nicht mehr als  $O(n)$  zusätzlichen Speicher brauchen.

13. algebraischen Spezifikation für Mengen, 7 Punkte

Leiten Sie aus der algebraischen Spezifikation für Mengen

$$\begin{aligned} \text{istenthalten}(x, \text{leereMenge}) &= \text{False} \\ \text{istenthalten}(x, \text{einfüge}(x, m)) &= \text{True} \\ \text{istenthalten}(x, \text{einfüge}(y, m)) &= \text{istenthalten}(x, m), & \text{für } x \neq y \\ \text{istenthalten}(x, \text{lösche}(x, m)) &= \text{False} \\ \text{istenthalten}(x, \text{lösche}(y, m)) &= \text{istenthalten}(x, m), & \text{für } x \neq y \end{aligned}$$

durch Umformungen folgende Identität her: Für  $x \neq y$  gilt

$$\text{istenthalten}(u, \text{lösche}(x, \text{einfüge}(y, m))) = \text{istenthalten}(u, \text{einfüge}(y, \text{lösche}(x, m))).$$

14. Vereinigung und Durchschnitt, 0 Punkte

Erweitern Sie die algebraische Spezifikation von Mengenoperationen aus der vorigen Aufgabe um die Funktionen *Vereinigung* und *Durchschnitt* (von je zwei Mengen).

---

<sup>1</sup><https://mycampus.imp.fu-berlin.de/access/content/group/fdc1e04e-407d-4d35-ac90-d6f03f853bc2/Programme/TopoSort.java>