IWI Fakultät für Informatik
und Wirtschaftsinformatik

Lehrgebiet für Grundlagen der Informatik
Prof. Dr. Heiko Körner

Hochschule Karlsruhe
Technik und Wirtschaft
UNIVERSITY OF APPLIED SCIENCES
Näher dran.

8. Übung zur Vorlesung Algorithmen auf Graphen

Musterlösungen

Aufgabe 1: Die Beweisschritte sind recht einfach zu zeigen:

a) Wenn eine starke Zusammenhangskomponente zwei Knoten u und v enthält, so existiert ein Pfad von u nach v und umgekehrt, d.h. die Komponente muss einen Kreis enthalten. Ein solcher Graph wäre dann aber nicht topologisch sortierbar.

b) Alle Knoten eines Kreises liegen immer in einer gemeinsamen starken Zusammenhangskomponente. Wenn also alle Komponenten trivial sind, kann es keinen einzigen solchen Kreis geben. Kreisfreie Graphen lassen sich jedoch immer topologisch sortieren.

c) Wenn jede starke Zusammenhangskomponente C nur aus einem einzigen Knoten v besteht, so fallen wegen

$$f(C) = \max\{f[w] \mid w \in C\} = \max\{f[w] \mid w \in \{v\}\} = f[v]$$

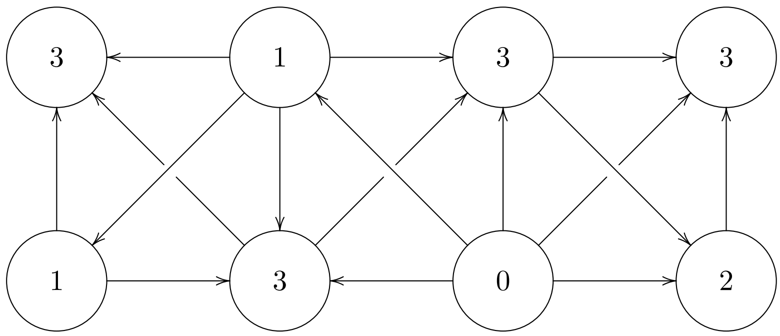
die Zeiten $f(C)$ und $f[v]$ zusammen. Seien nun v und v' zwei beliebige Knoten, die durch eine Kante von v nach v' verbunden sind und in den trivialen Komponenten $\{v\}$ und $\{v'\}$ liegen. Im Satz 1.3.9 der Vorlesung wurde gezeigt, dass nach der ersten Phase dann stets $f(\{v\}) > f(\{v'\})$, also $f[v] > f[v']$ gilt. Weiterhin liegen nach der ersten Phase alle Knoten absteigend nach ihren f–Werten sortiert auf dem Keller L vor. Also wird der Knoten v vor dem Knoten v' als einzelne Komponente ausgegeben.

Da diese Argumentation für alle Knoten v, v' mit $(v, v') \in E$ greift, muss die Ausgabe insgesamt einer topologischen Sortierung entsprechen. $\square$

Aufgabe 2: Hier die Musterlösungen zu beiden Teilaufgaben:

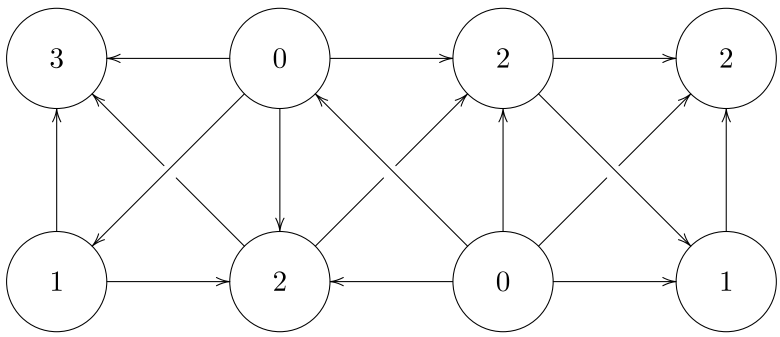
a) In den ersten neun Zeilen des Algorithmus Topologische Sortierung werden die *indeg*–Werte so initialisiert, dass sie den Eingangsgraden der einzelnen Knoten entsprechen. Wir notieren diese Werte wieder in den Knotenkreisen:

$Q = \emptyset$	Bislang erfolgte Ausgabe: —



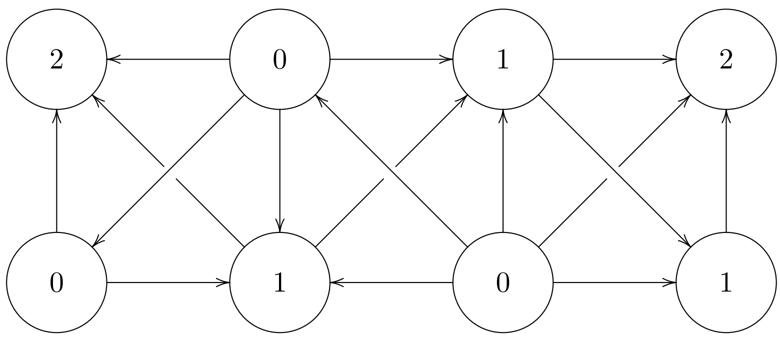
Q enthält nach der Ausführung der Zeilen 10–14 nur den Knoten g, welcher anschließend in Zeile 16 ausgewählt, ausgegeben, und aus Q wieder entfernt wird. Für alle Nachfolgerknoten b, c, d, f und h wird danach (Zeilen 19–24) der *indeg*–Wert einmal reduziert. Der einzige Knoten mit einem *indeg*–Wert von 0 ist der Knoten b, welcher dann also auch das einzige Element von Q darstellt:

$Q = \{b\}$	Bislang erfolgte Ausgabe: (g)



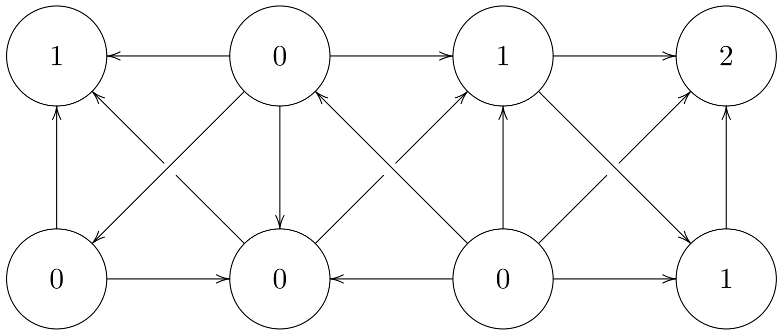
Die Verarbeitung von b verringert die *indeg*–Werte von a, c, e und f und führt zu einer Aufnahme von e nach Q:

$Q = \{e\}$	Bislang erfolgte Ausgabe: (g, b)



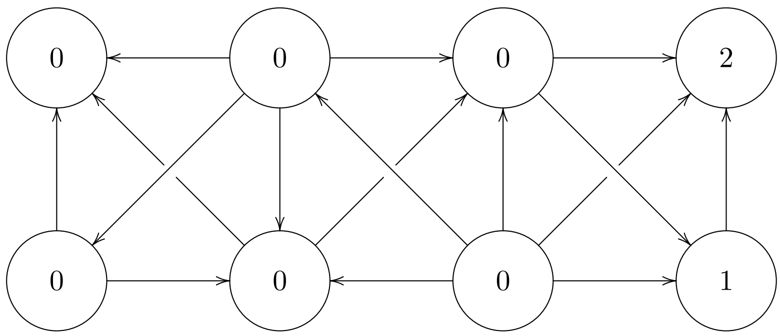
Auch hier ist der nächste Verarbeitungsschritt eindeutig. Knoten e verringert die *indeg*–Werte von a und f, womit f dann den Wert 0 erreicht:

$Q = \{f\}$	Bislang erfolgte Ausgabe: (g, b, e)



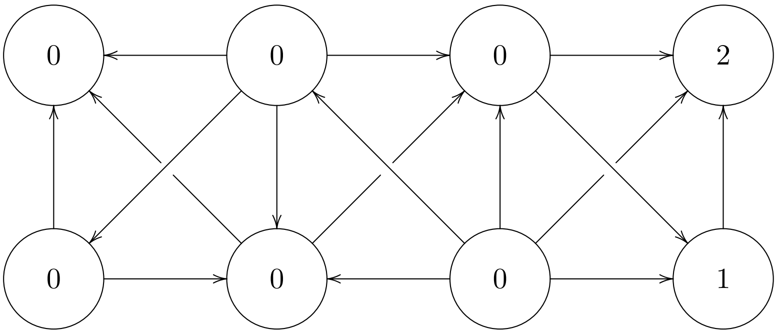
Durch die Verarbeitung von f werden die *indeg*–Werte von sowohl a als auch c auf 0 gesetzt:

$Q = \{a, c\}$	Bislang erfolgte Ausgabe: (g, b, e, f)



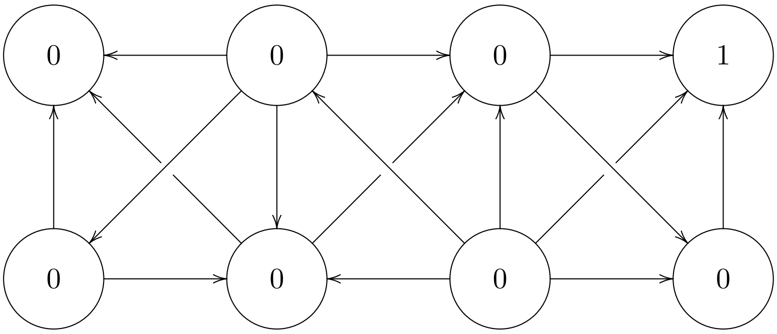
Erstmals in diesem Beispiel besteht für die weitere Verarbeitung die Wahl zwischen zwei Knoten. Falls z.B. a in Zeile 16 ausgewählt wird, so hat dies (abgesehen von der Ausgabe dieses Knotens) keinen Effekt, da a keine ausgehenden Kanten besitzt:

$Q = \{c\}$	Bislang erfolgte Ausgabe: (g, b, e, f, a)



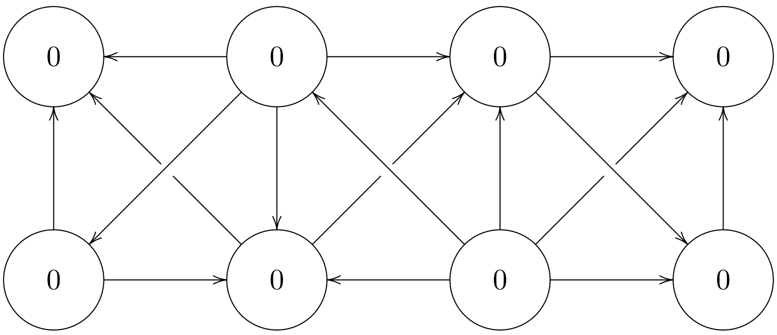
Die weitere Verarbeitung ist nun wieder eindeutig. Der Knoten c verringert die *indeg*–Werte von d und h:

$Q = \{h\}$	Bislang erfolgte Ausgabe: (g, b, e, f, a, c)



Die Verarbeitung von h verringert anschließend den *indeg*–Wert von d:

$Q = \{d\}$	Bislang erfolgte Ausgabe: (g, b, e, f, a, c, h)



Da d keine ausgehenden Kanten besitzt, bleibt die abschließende Verarbeitung dieses Knotens ohne Wirkung:

$Q = \emptyset$	Bislang erfolgte Ausgabe: (g, b, e, f, a, c, h, d)

Wegen $Q = \emptyset$ endet an dieser Stelle der Algorithmus. Die Ausgabe ist eine korrekte topologische Sortierung aller Knoten.

b) Wenn man den obigen Programmablauf noch einmal genau nachvollzieht, so sieht man, dass die ermittelte topologische Sortierung weitestgehend eindeutig ist. Lediglich die Ausgabe des Knotens a kann — sobald er den *indeg*–Wert 0 besitzt, also nach der Ausgabe des Knotens f — beliebig nach hinten verzögert werden, da a selbst keine Nachfolgerknoten besitzt. Folglich gibt es die folgenden vier möglichen topologische Sortierungen:

$$(g, b, e, f, a, c, h, d) \ , \quad (g, b, e, f, c, a, h, d) \ , \quad (g, b, e, f, c, h, a, d) \ , \quad (g, b, e, f, c, h, d, a) \ .$$

Aufgabe 3: Bzgl. der topologischen Sortierungen lauten die Ergebnisse wie folgt:

a) G enthält mehrere Zyklen. Eine topologische Sortierung ist jedoch nur bei azyklischen Graphen möglich.

b) Der Algorithmus Topologische Sortierung würde mit dem Knoten e beginnen, da e der einzige Knoten mit Eingangsgrad 0 ist. Nach der Ausgabe von e würde der Algorithmus aber auch direkt wieder stoppen, denn alle anderen Knoten sind entweder auf Kreisen gebunden oder von solchen Kreisknoten abhängig.

c) Es handelt sich um die Kante (d, g). Nach der Entnahme dieser Kante sind in G keine Kreise mehr vorhanden, und eine topologische Sortierung wird ermöglicht. Entfernt man irgend eine andere Kante, so bleibt zumindest ein Kreis erhalten, so dass G nach wie vor nicht topologisch sortiert werden kann.

d) Eine mögliche topologische Sortierung ist (e, g, h, b, f, c, d, a), und es gibt noch viele weitere mögliche Lösungen.

Aufgabe 4: Bzgl. der topologischen Sortierungen lauten die Ergebnisse wie folgt:

a) G enthält zwei Zyklen. Eine topologische Sortierung ist jedoch nur bei azyklischen Graphen möglich. Wenn aber die Kante (h, d) entfernt wird, so werden beide Zyklen zerstört.

b) Insgesamt gibt es acht verschiedene topologische Sortierungen (eine davon ist für eine korrekte Lösung der Aufgabe ausreichend), nämlich

$$(b, d, c, g, h, f, e, a) \ , \quad (d, b, c, g, h, f, e, a) \ , \quad (b, d, c, g, f, h, e, a) \ , \quad (d, b, c, g, f, h, e, a)$$

und

$$(b, d, c, g, f, e, h, a) \ , \quad (d, b, c, g, f, e, h, a) \ , \quad (b, d, c, g, f, e, a, h) \ , \quad (d, b, c, g, f, e, a, h) \ .$$

Aufgabe 5: Bzgl. der topologischen Sortierungen lauten die Ergebnisse wie folgt:

a) G enthält mehrere Zyklen. Eine topologische Sortierung ist jedoch nur bei azyklischen Graphen möglich.

b) Der Algorithmus Topologische Sortierung würde nur g und dann f ausgeben, denn alle anderen Knoten sind auf Kreisen gebunden.

c) Man muss die ausgehenden Kanten von c entfernen, denn nur dieser Knoten liegt auf beiden Kreisen. Entfernt man die ausgehenden Kanten eines anderen Knotens, bleibt zumindest ein Kreis erhalten, so dass G nach wie vor topologisch nicht sortiert werden kann.

d) Eine mögliche topologische Sortierung ist z.B. (g, f, h, b, a, d, e, c), und es gibt noch weitere Möglichkeiten:

- Allen Sortierungen ist die Teilsequenz (g, f, b, a, e, c) gemein.
- Der Knoten h taucht zwischen g und c auf.
- Der Knoten d muss sich zwischen h und c befinden.

Insgesamt ergeben sich so $1 + 2 + 3 + 4 + 5 = 15$ Möglichkeiten.