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.

# 4. Übung zur Vorlesung Algorithmen auf Graphen

## Musterlösungen

**Aufgabe 1:** Diese Aufgabe hat die folgenden Lösungen:

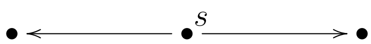
a) Der Knoten $g$ ist nur von sich selbst aus erreichbar. Wenn man also von irgendeinem anderen Knoten mit dem DFS–Verfahren starten würde, so würde $g$ nicht markiert. Daher würde $g$ auch nie auf dem DFS–Keller landen, und somit könnten insbesondere nie alle acht Knoten gleichzeitig im Keller stehen. Als Startknoten kommt deshalb nur $g$ in Frage.

b) Die einzige mögliche Folge ist $g, f, b, a, e, c, h, d$.

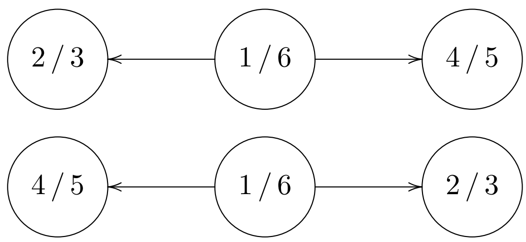
**Aufgabe 2:** Nach dem Auflegen von $v$ auf den Keller zum Zeitpunkt $d[v]$ vergehen $f[v] - d[v]$ viele Schritte bis zum Entfernen von $v$ zum Zeitpunkt $f[v]$. Wenn man also das Entfernen von $v$ nicht mitzählt, so vergehen zwischen dem Speichern und dem Entnehmen von $v$ genau $f[v] - d[v] - 1$ viele Schritte. Diese Zahl muss jedoch gerade sein (und damit ist $f[v] - d[v]$ dann ungerade). Denn jeder Knoten, der zwischenzeitlich auf den Keller gelangt, muss auch vor dem Entfernen von $v$ wieder gelöscht werden, d.h. für jeden solchen Knoten wird die Uhr um insgesamt zwei Schritte und damit um einen gerade Zahl weitergezählt. □

**Aufgabe 3:** In den nachstehenden Durchmusterungsergebnissen sind für jeden Knoten $v$ die *Push*– und *Pop*–Zeiten als Zahlenpaare $(d[v] \,/\, f[v])$ eingetragen.

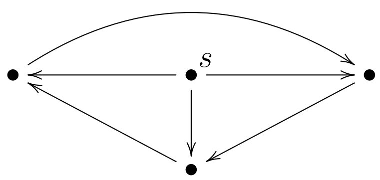
a) Hier reicht der einfache Graph




Die beiden möglichen Durchmusterungen sind dann diese:

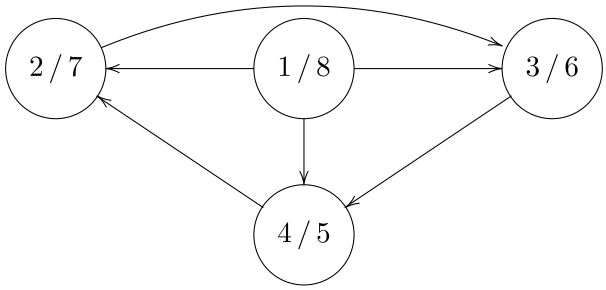


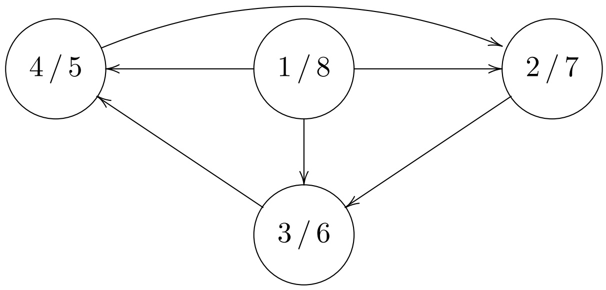

b) Ein mögliche Lösung wäre dieser Graph:

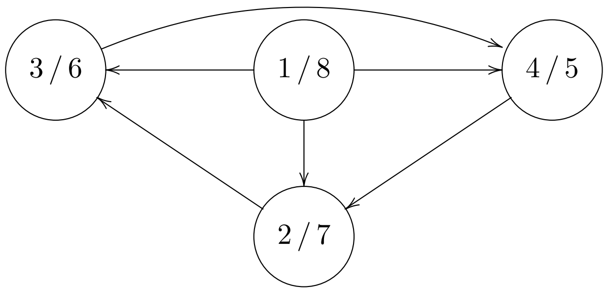



Sprechstunde Di 10-11 Raum E 204 ∗ Tel.: (0721) 925–1507 ∗ Email: heiko.koerner[at]hs–karlsruhe[dot]de

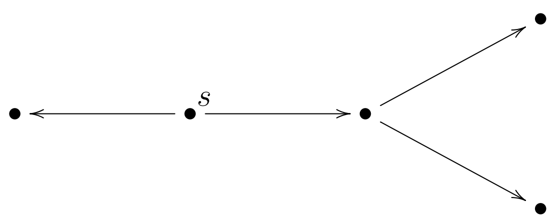
Die drei möglichen DFS–Durchmusterungen lauten dann:



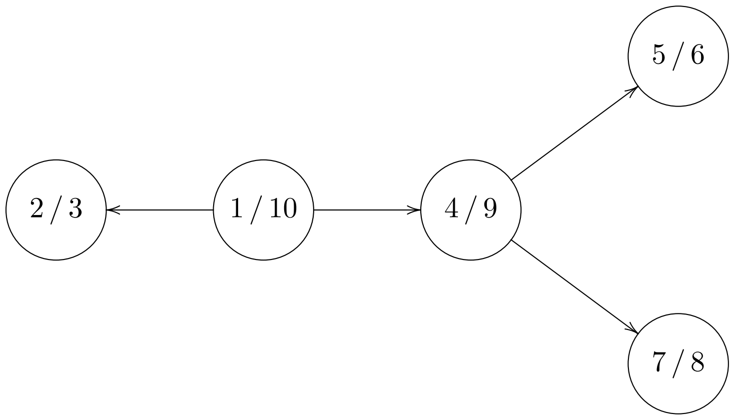



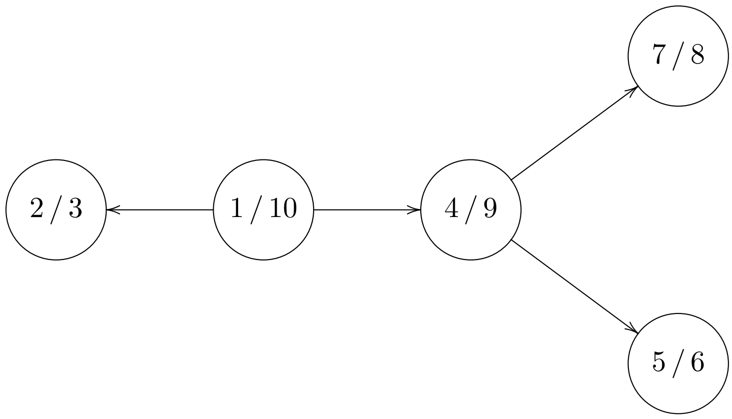



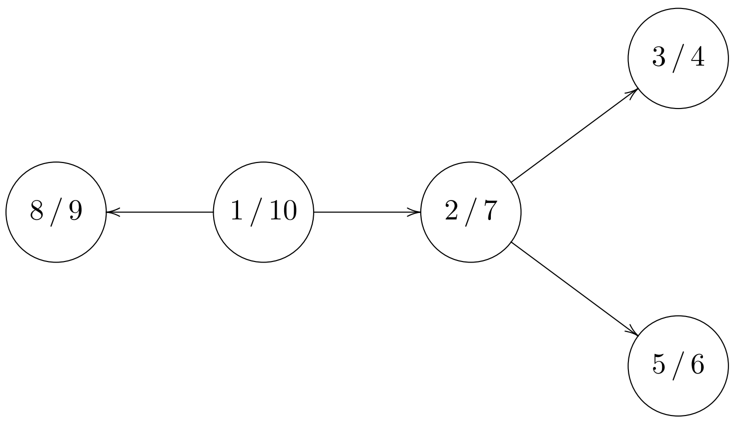

c) Ein Graph mit vier möglichen DFS–Durchmusterungen ist dieser:

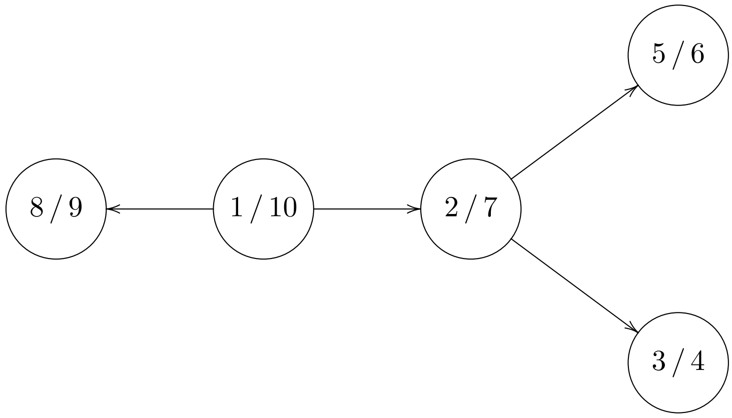



Wir erhalten diese vier möglichen DFS–Ergebnisse:


5 / 6
2 / 3
1 / 10
4 / 9
7 / 8


7 / 8
2 / 3
1 / 10
4 / 9
5 / 6


3 / 4
8 / 9
1 / 10
2 / 7
5 / 6




**Aufgabe 4:** Wir geben die Konfigurationen an, die jeweils beim Erreichen des Kopfes der `while`–Schleife in Zeile 5 gelten.

—

$\Downarrow$ Die ersten vier Zeilen werden durchlaufen

$$\boxed{Q = (a) \qquad N[a] = \{e, f\}}$$

$\Downarrow$ In Zeile 6 gilt $head(Q) = a$; wähle z.B. $e \in N[a]$ in Zeile 8

$$\boxed{Q = (a, e) \qquad N[a] = \{f\} \qquad N[e] = \{a, f\}}$$

$\Downarrow$ In Zeile 6 gilt $head(Q) = a$; wähle (eindeutig) $f \in N[a]$ in Zeile 8

$$\boxed{Q = (a, e, f) \qquad N[a] = \emptyset \qquad N[e] = \{a, f\} \qquad N[f] = \{g\}}$$

$\Downarrow$ $a$ wird wegen $N[a] = \emptyset$ in Zeile 16 aus $Q$ entfernt

$$\boxed{Q = (e, f) \qquad N[e] = \{a, f\} \qquad N[f] = \{g\}}$$

$\Downarrow$ In Zeile 6 gilt $head(Q) = e$; wähle z.B. markiertes $a \in N[e]$ in Zeile 8

$$\boxed{Q = (e, f) \qquad N[e] = \{f\} \qquad N[f] = \{g\}}$$

$\Downarrow$ In Zeile 6 gilt $head(Q) = e$; wähle markiertes $f \in N[e]$ in Zeile 8

$$\boxed{Q = (e, f) \qquad N[e] = \emptyset \qquad N[f] = \{g\}}$$

$\Downarrow$ $e$ wird wegen $N[e] = \emptyset$ in Zeile 16 aus $Q$ entfernt

$$\boxed{Q = (f) \qquad N[f] = \{g\}}$$

$\Downarrow$ In Zeile 6 gilt $head(Q) = f$; wähle (eindeutig) $g \in N[f]$ in Zeile 8

$$\boxed{Q = (f, g) \qquad N[f] = \emptyset \qquad N[g] = \{d\}}$$

$\Downarrow$ $f$ wird wegen $N[f] = \emptyset$ in Zeile 16 aus $Q$ entfernt

$$Q = (g) \qquad N[g] = \{d\}$$

$\Downarrow$ In Zeile 6 gilt $head(Q) = g$; wähle (eindeutig) $d \in N[g]$ in Zeile 8

$$Q = (g, d) \qquad N[g] = \emptyset \qquad N[d] = \{h\}$$

$\Downarrow$ $g$ wird wegen $N[f] = \emptyset$ in Zeile 16 aus $Q$ entfernt

$$Q = (d) \qquad N[d] = \{h\}$$

$\Downarrow$ In Zeile 6 gilt $head(Q) = d$; wähle (eindeutig) $h \in N[d]$ in Zeile 8

$$Q = (d, h) \qquad N[d] = \emptyset \qquad N[h] = \{g\}$$

$\Downarrow$ $d$ wird wegen $N[d] = \emptyset$ in Zeile 16 aus $Q$ entfernt

$$Q = (h) \qquad N[h] = \{g\}$$

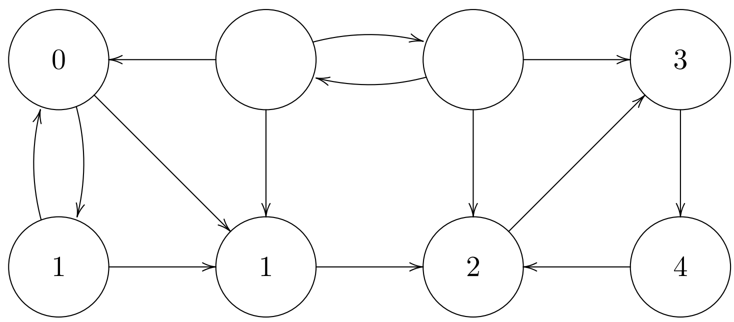
$\Downarrow$ In Zeile 6 gilt $head(Q) = h$; wähle markiertes $g \in N[h]$ in Zeile 8

$$Q = (h) \qquad N[h] = \emptyset$$

$\Downarrow$ $h$ wird wegen $N[h] = \emptyset$ in Zeile 16 aus $Q$ entfernt

$$Q = ()$$

Wegen der Leere von $Q$ terminiert der Algorithmus dann an dieser Stelle. Die von BFS ermittelten Attribute sind in dem folgenden Graph eingetragen:




Die Knotenattribute geben wie erwartet die kürzesten Entfernungen (d.h. die Anzahl der Kanten) auf Pfaden von $a$ aus an.

**Aufgabe 5:** Die Lösungen zu der geforderte Analyse von BFS sind:

a) Von $e$ aus sind alle acht Knoten des Graphen erreichbar. Die von BFS ermittelten Distanzen lauten:

| Knoten $v$ | $a$ | $b$ | $c$ | $d$ | $e$ | $f$ | $g$ | $h$ |
|---|---|---|---|---|---|---|---|---|
| Distanz $d[v]$ | 1 | 1 | 2 | 3 | 0 | 1 | 4 | 5 |

b) Es handelt sich um die Knoten $a$ (Maximalwert 0), $f$ (Maximalwert 1) und $g$ (Maximalwert 2).