Aufgabe 1

Für jeden Knoten $u \in V$ speichern wir alle Kanten $[u, v\rangle$ in einer Hashtabelle, wobei die Tabelle selbst eine Länge in $b \times \text{outdeg}(u)$ für eine Konstante b besitzt und Konflikte mit Verkettung gelöst werden. Wenn die Hashfunktion zufällig aus einer c-universellen Familie gewählt wird, so ist die erwarte Länge der Kollisionslisten konstant.

Für die Ausgabe aller von u ausgehenden Kanten kann man durch die Hashtabelle und durch alle Kollisionslisten in Zeit $\Theta(\text{outdeg}(u))$ iterieren.

Um zu testen ob eine gegebene Kante $[u, v\rangle$ im Graphen enthalten ist, sucht man in der Hashtabelle von u durch die jeweilige Kollisionsliste. Da die erwartete Länge der Kollisionsliste konstant ist, erfolgt der Test in erwarteter konstanter Laufzeit.

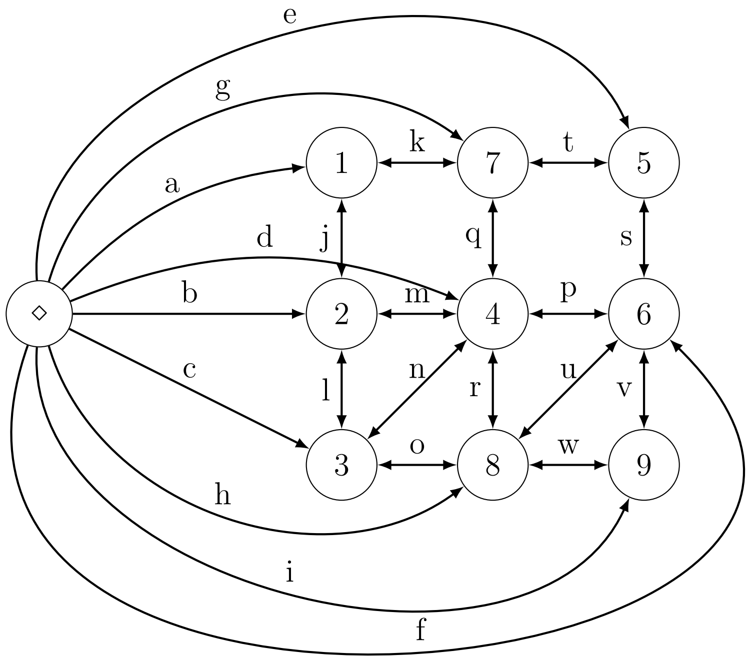
Die Größe der Datenstruktur für die reinen Tabellen ohne Kollisionslisten ist $b|E|$, da jede Kante genau eine Tabelle um b vergrößert. Die Größe aller zusätzlichen Felder für die Kollisionslisten ist durch $|E|$ beschränkt, da jede Kante nur in einer Kollisionsliste auftaucht aber jede Stelle ein Element enthält. Die gesamtgröße der Datenstruktur ist also $(b + 1)|E| + |V|$, proportional zur Größe des Graphen.

Aufgabe 3

- $\Rightarrow$ Existiert eine Kante, die von einem Knoten zu einem seiner Ahnen im lex-kleinsten-Wege-Baum führt, so bildet diese Kante mit dem gerichteten Baumpfad von dem Ahnen zum Knoten einen gerichteten Zyklus.
- $\Leftarrow$ Existiert ein gerichteter Zyklus, so lässt sich der Knoten u im Zyklus bestimmen, der im lex-kleinsten-Wege-Baum den geringsten Rang besitzt. Der Vorgänger von u im Zyklus ist ein Nachfahre von u im lex-kleinsten-Wege-Baum, da er von u aus durch den Zyklus erreichbar ist und nicht bereits im lex-kleinsten-Wege-Baum auftaucht, da u der rangniedrigste Knoten aus dem Zyklus ist. Folglich ist die Kante die im Zyklus nach u führt eine Kante die von einem Knoten zu einem seiner Ahnen im lex-kleinsten-Wege-Baum führt.

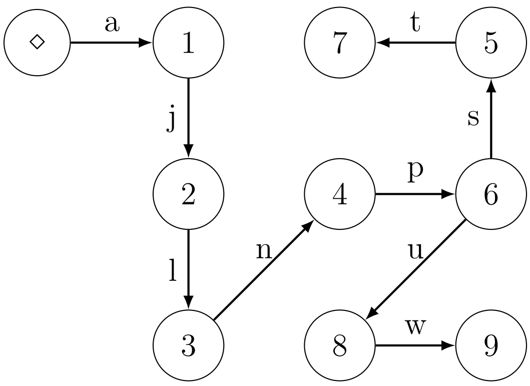
Aufgabe 2

(a)



(b)

Knoten	Rang	$d[\cdot]$	$\pi[\cdot]$	$f[\cdot]$
◇	0	0	⊥	19
1	1	1	◇	18
2	2	2	1	17
3	3	3	2	16
4	4	4	3	15
6	5	5	4	14
5	6	6	6	9
7	7	7	5	8
8	8	10	6	13
9	9	11	8	12



Rang und d, f-Nummerierung lassen wir hier im augmentierten Graphen bei 0 beginnen, da die Werte dadurch den entsprechenden üblichen Werten im nicht augmentierten Graphen entsprechen.