Lösung zu Aufgabe 4

Teil (a)

Zu zeigen ist die Korrektheit der Invariante für die Hauptschleife des Algorithmus.

$$\texttt{EGrad}_i[u] = \texttt{EGrad}_0[u] - \sum_{j=1}^{i} A[v[j], u] \tag{1}$$

Für $i = 0$, also vor Betreten der Schleife ist die Invariante erfüllt, denn:

$$\texttt{EGrad}_0[u] = \texttt{EGrad}_0[u] - \underbrace{\sum_{j=1}^{0} A[v[j], u]}_{=0} \quad \Leftrightarrow \quad \texttt{EGrad}_0[u] = \texttt{EGrad}_0[u] \tag{2}$$

Wir interessieren uns im Folgenden nur für vollständige Schleifendurchläufe. Es kann folglich in jedem der betrachteten Fälle ein Knoten aus der Schlange entnommen werden.

Für $i = 1$ wird der Knoten $v[1]$ aus der Schlange entnommen und für jeden Knoten aus der Adjazenzliste von $v[1]$ der Eingangsgrad um eins verringert. Da jeder Knoten in der Adjazenzliste nur einmal vorkommen kann, wird der Eingangsgrad eines bestimmten Knotens auch nur einmal verändert. Der Eingangsgrad eines bestimmten Knotens u wird genau dann um eins kleiner, wenn der Knoten $v[1]$ auf u zeigt. Das entspricht genau dem Fall, dass $A[v[1], u] = 1$ ist. Es gilt:

$$\begin{aligned} & \texttt{EGrad}_1[u] = \texttt{EGrad}_0[u] - A[v[1], u] \\ \Leftrightarrow \quad & \texttt{EGrad}_1[u] = \texttt{EGrad}_0[u] - \underbrace{\sum_{j=1}^{1} A[v[j], u]}_{=A[v[1],u]} \end{aligned} \tag{3}$$

Dies können wir als Induktionsvoraussetzung benutzen. Nehmen wir an, die Behauptung gilt für $i = k$. Für $k = 1$ ist das nach obiger Überlegung der Fall. Also gilt für $k \leq 1$:

$$\texttt{EGrad}_k[u] = \texttt{EGrad}_0[u] - \sum_{j=1}^{k} A[v[j], u] \tag{4}$$

Jetzt machen wir den Schritt von $i = k$ nach $i = k + 1$. Im Schleifendurchlauf $k + 1$ wird der Knoten $v[k+1]$ aus der Schlange entnommen. Analog zu obiger Überlegung ergibt sich im Schritt 5 des Algorithmus der Eingangsgrad eines Knotens u als:

$$\texttt{EGrad}_{k+1}[u] = \texttt{EGrad}_k[u] - A[v[k + 1]] \tag{5}$$

Setzen wir jetzt für $\texttt{EGrad}_k[u]$ die Induktionsvoraussetzung ein, erhalten wir:

$$\begin{aligned} & \texttt{EGrad}_{k+1}[u] = \texttt{EGrad}_0[u] - \sum_{j=1}^{k} A[v[j],u] - A[v[k+1]] \\ \Leftrightarrow\quad & \texttt{EGrad}_{k+1}[u] = \texttt{EGrad}_0[u] - \sum_{j=1}^{k+1} A[v[j],u] \end{aligned} \tag{6}$$

Somit gilt die Behauptung auch, wenn die Schleife nach dem k-ten mal ein $(k+1)$-tes mal durchlaufen wird. Da die Behauptung für $k = 0$ und auch für $k = 1$ gilt, gilt sie folglich solange bis die Schleife abbricht. $\square$

Teil (b)

Zu zeigen ist: Aus n Schleifendurchläufen folgt, dass $(v[1], ..., v[n])$ eine topologische Sortierung des Graphen ist.

Die Schleife ist komplett durchgelaufen und $v[1], ..., v[n]$ sind die im jeweiligen Durchlauf entfernten Knoten.

Nehmen wir einmal an, die $(v[1], ..., v[n])$ bilden *keine* topologische Sortierung. Dann muss es zwei Knoten $v[k]$ und $v[m]$ mit $k < m \leq n$ und eine Kante $(v[m], v[k])$ geben, die die topologische Sortierung verletzt.

Da der Knoten $v[k]$ im k-ten Schritt aus der Schlange geholt wird, muss er spätestens im Schritt $k-1$ eingefügt worden sein. Nehmen wir an im Schritt $l \leq k-1$. Das passiert nur dann, wenn gilt:

$$\texttt{EGrad}_l[v[k]] = \texttt{EGrad}_0[v[k]] - \sum_{j=1}^{l} A[v[j], v[k]] = 0 \tag{7}$$

Betrachten wir jetzt die Invariante genauer. Wir wissen, es gab n Druchläufe der Schleife. Also gilt:

$$\begin{aligned} \texttt{EGrad}_n[v[k]] &= \texttt{EGrad}_0[v[k]] - \sum_{j=1}^{n} A[v[j], v[k]] \\ &= \underbrace{\texttt{EGrad}_0[v[k]] - \sum_{j=1}^{l} A[v[j], v[k]]}_{=\texttt{EGrad}_l[v[k]]=0} - \sum_{j=l+1}^{n} A[v[j], v[k]] \end{aligned} \tag{8}$$

In der Adjazenzmatrix kommen nur die Zahlen 0 und 1 vor. Also muss $\sum_{j=l+1}^{n} A[v[j], v[k]] = 0$ gelten. Insbesondere muss $A[v[m], v[k]] = 0$ sein, im Widerspruch zur Annahme eine solche Kante existiert. Also muss es sich doch um eine topologische Sortierung handeln. $\square$

Teil(c)

Zu zeigen ist: Wenn der Algorithmus „Kreis“ ausgibt, ist auch tatsächlich ein Kreis vorhanden.

Die Schleife wurde irgendwann nach k Schritten mit einer leeren Schlange abgebrochen. Die Knoten $v[1], ..., v[k]$ sind die bis dahin entfernten Knoten.

Nehmen wir an, der Graph enthält doch *keinen* Kreis. Dann gibt es eine topologische Sortierung des Graphen, die mit $v[1], ..., v[k]$ beginnt. Die restlichen Knoten der topologischen Sortierung seien mit $v[k+1], ..., v[n]$ bezeichnet.

Da der Algorithmus nach $k < n$ Schritten angehalten hat, wurde irgendwann nichts mehr in die Schlange eingefügt. Die letzte Möglichkeit, einen neuen Knoten in die Schlange einzufügen, war im Schritt k. Da dies *nicht* passiert ist, muss für alle restlichen Knoten $u \in \{v[k+1], ..., v[n]\}$ $\texttt{EGrad}_k[u] > 0$ gelten. Insbesondere gilt für $v[k+1]$:

$$\texttt{EGrad}_k[v[k+1]] = \texttt{EGrad}_0[v[k+1]] - \sum_{j=1}^{k} A[v[j], v[k+1]] > 0 \tag{9}$$

Es muss folglich noch mindestens einen Eintrag $A[v[m], v[k+1]] = 1$ mit $m > k$ geben, also eine Kante von einem der $v[k+2], ..., v[n]$ zu $v[k]$. Dies steht im Widerspruch zur Annahme $(v[1], ..., v[n])$ ist eine topologische Sortierung des Graphen.

Einen Kreis findet man, indem man die eingehenden Kanten der restlichen Knoten rückwärts verfolgt. Es gibt $n-k$ Knoten und jeder hat mindestens eine eingehende Kante. Jeder Knoten kann somit wieder verlassen werden, aber wir können höchstens $n-k-1$ mal zu einem neuen Knoten kommen. Sobald ein bereits besuchter Knoten getroffen wird, schließt sich der Kreis. $\square$

Teil (d)

Den Fall $\texttt{AGrad} = 0$ kann man in Adjazenzlistendarstellung direkt an einer leeren Liste erkennen. Es entfällt damit das Zählen am Anfang des Algorithmus. Allerdings müssen jetzt beim Löschen eines Knotens alle Adjazenzlisten der verbleibenden Knoten abgesucht werden. Ein Vorteil bei der Laufzeit ist dadurch ohne Modifikation der Datenstruktur eher nicht zu erwarten.