



Hausaufgaben zur Vorlesung »Analyse von Algorithmen und Komplexität«

Blatt 9

Definition 1 (Vertex Cover (VC)). Für einen (ungerichteten) Graphen $G = (V, E)$ ist ein Vertex Cover eine Menge von Knoten $C \subseteq V$, sodass $u \in C$ oder $v \in C$ für alle $\{u, v\} \in E$ gilt, also ist für alle Kanten mindestens ein Endpunkt in C .

Für das VERTEXCOVER Problem ist neben einem Graphen G eine Zahl $k \in \mathbb{Z}_{\geq 0}$ gegeben und es wird gefragt, ob ein Vertex Cover mit Größe (Anzahl der Knoten) höchstens k in G existiert.

Definition 2 (Feedback Vertex Set (FVS)). Für einen gerichteten Graphen $G = (V, E)$ ist ein Feedback Vertex Set eine Menge von Knoten $X \subseteq V$, sodass $G \setminus X$ kreisfrei ist.

Für das FEEDBACKVERTEXSET Problem ist neben einem Graphen G eine Zahl $k \in \mathbb{Z}_{\geq 0}$ gegeben und es wird gefragt, ob ein Feedback Vertex Set mit Größe (Anzahl der Knoten) höchstens k in G existiert.

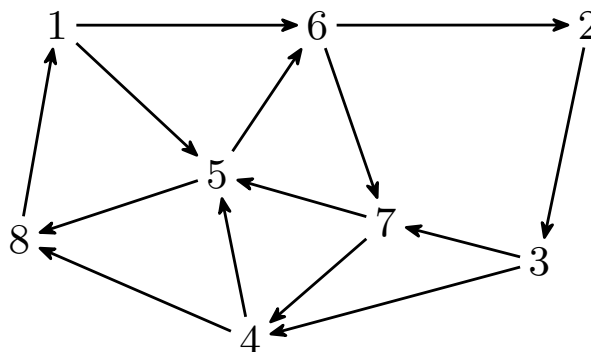


Abbildung 1: Beispiel Graph.

Hausaufgabe 9.1 (Nichtdeterministischer Algorithmus für VC)

(5 Punkte)

- Finden Sie ein minimales Vertex Cover in Abbildung 1. Für ein Vertex Cover ist die Orientierung der Kanten nicht relevant.
- Zeigen Sie $\text{VERTEXCOVER} \in \text{NP}$, indem Sie einen nichtdeterministischen Algorithmus angeben, der VERTEXCOVER in Polynomialzeit löst. Begründen Sie die Korrektheit und die Laufzeit von Ihrem Algorithmus.

Lösung. 1. Ein minimales Vertex Cover ist zum Beispiel $\{3, 5, 6, 7, 8\}$.

2. Unser nichtdeterministischer Algorithmus geht wie folgt vor: Für jeden Knoten wird nichtdeterministisch entschieden, ob dieser im Vertex Cover sein soll. Nachdem dies für alle Knoten entschieden wurde, wird überprüft, ob nicht mehr als k Knoten gewählt wurden und ob für jede Kante mindestens ein Endpunkt im Vertex Cover liegt.

Existiert ein Vertex Cover im Graphen, existiert auch eine Reihe von nichtdeterministischen Entscheidungen, die diese Teilmenge von Knoten auswählt. Diese Teilmenge wird dann auch von dem Algorithmus akzeptiert. Damit ist der Algorithmus korrekt.

Beide diese Schritte können naiv in Polynomialzeit erledigt werden, also löst dieser Algorithmus das Vertex Cover Problem in Polynomialzeit. $\square$

Bewertung 1 Punkt für das Beispiel und 4 Punkte für den Algorithmus, dabei 2 Punkte für die Beschreibung und jeweils 1 Punkt für Korrektheit und Laufzeit.

Hausaufgabe 9.2 (Polynomieller Verifizierer für FVS)

(5 Punkte)

1. Finden Sie ein minimales Feedback Vertex Set in Abbildung 1.
2. Zeigen Sie $\text{FEEDBACKVERTEXSET} \in \mathbf{NP}$, indem Sie einen polynomiellen Verifizierer für FEEDBACKVERTEXSET angeben. Begründen Sie die Korrektheit und die Laufzeit von Ihrem Verifizierer.

Lösung. 1. Ein minimales FVS ist zum Beispiel $\{5, 6\}$.

2. Das Zertifikat hat die Form einer Teilmenge $X \subseteq V$, die beschreibt, welche Knoten das Feedback Vertex Set bilden sollen. Zum Beispiel über ein Bit pro Knoten. Nun wird überprüft, ob $|X| \leq k$ gilt und ob $G \setminus X$ azyklisch ist.

Hat der Graph ein Feedback Vertex Set von Größe maximal k , gibt es ein Zertifikat, welches dieses beschreibt. Dieses Zertifikat wird dann von unserem Verifizierer akzeptiert, damit ist der Verifizierer korrekt.

Die Größe von X zu überprüfen ist naiv in Polynomialzeit möglich. Für den zweiten Test bietet sich der Algorithmus zur topologischen Sortierung aus der Vorlesung an, welcher in Polynomialzeit eine topologische Sortierung findet, wenn der gegebene Graph azyklisch ist. Insgesamt ist dieser Verifizierer korrekt und auch ein polynomieller Verifizierer.

Bewertung 1 Punkt für das Beispiel und 4 Punkte für den Verifizierer, dabei 2 Punkte für die Beschreibung und jeweils 1 Punkt für Korrektheit und Laufzeit. $\square$