Übungen zur Vorlesung Datenstrukturen und Algorithmen

T22

Wie kann sich die Anzahl der starken Zusammenhangskomponenten eines gerichteten Graphen ändern, wenn wir eine einzige Kante hinzufügen?

Lösungsvorschlag:

Es ist leicht einzusehen, daß sich diese Anzahl in bestimmten Fällen nicht verändert oder um eins sinkt. Betrachte aber etwa einen Graphen, der lediglich aus einem gerichteten Pfad mit n Knoten besteht. Dieser Graph hat n starke Zusammenhangskomponenten, die durch Kreisschluß mittels einer zusätzlichen Kante vom Blatt zur Wurzel zu einer Komponente zusammenfallen. Wenn wir die zusätzliche Kante weiter oben im Baum anfangen lassen, können wir sogar jede beliebige Anzahl von starken Zusammenhangskomponenten zwischen 2 und $n-1$ erzeugen.

T23

Beweisen oder widerlegen Sie: Wenn man einen ungerichteten Graphen $G = (V, E)$ in einen gerichteten Graphen $G' = (V, E')$ umwandelt, indem man jede ungerichtete Kante $\{u, v\}$ durch zwei gerichtete Kanten (u, v) und (v, u) ersetzt, dann ist G genau dann kreisfrei, wenn eine Tiefensuche auf G' keine Vorwärtskante zutage fördert.

Lösungsvorschlag:

Die Aussage ist wahr: Wenn es in G' eine Vorwärtskante (u, v) gibt, dann gibt es auch einen Pfad aus Baumkanten von u nach v, der mindestens zwei Kanten lang ist. In G bilden dieser Pfad und die Kante $\{u, v\}$ einen Kreis (mindestens der Länge drei).

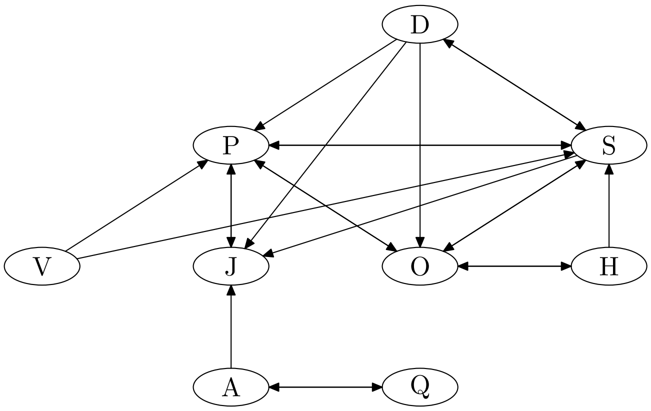
Nehmen wir umgekehrt an, es gibt einen Kreis in G. Dann gibt es auch einen gerichteten Kreis in G'. Sei v der Kreisknoten mit der kleinsten discovery time. Dann ist die Kreiskante e, die auf v zurückführt, eine Rückwärtskante: Alle Knoten auf dem Kreis liegen in der gleichen starken Zusammenhangskomponente, so daß e keine Querkante sein kann. Sie kann auch keine Baumkante sein, da v von einem anderen Knoten aus zuerst erreicht wurde. Schließlich kann e keine Vorwärtskante sein, weil die discovery time ihres Startknotens größer ist als die von v. Der Rückkante von e steht natürlich eine Vorwärtskante gegenüber.

T24

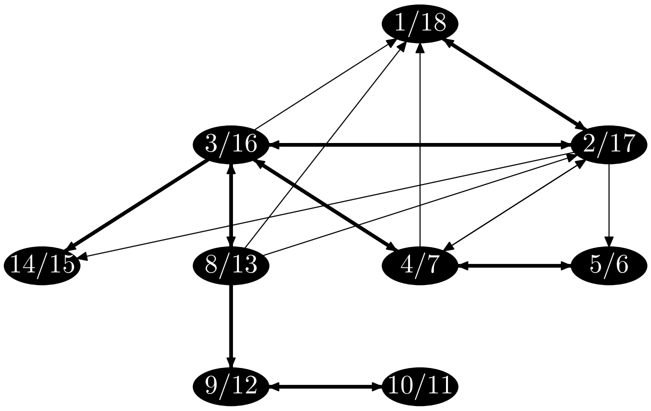
Beweisen oder widerlegen Sie die folgende Aussage: Es gibt einen Algorithmus, der ungerichtete Graphen $G = (V, E)$ in Zeit $O(|V|)$ auf Azyklizität prüft.

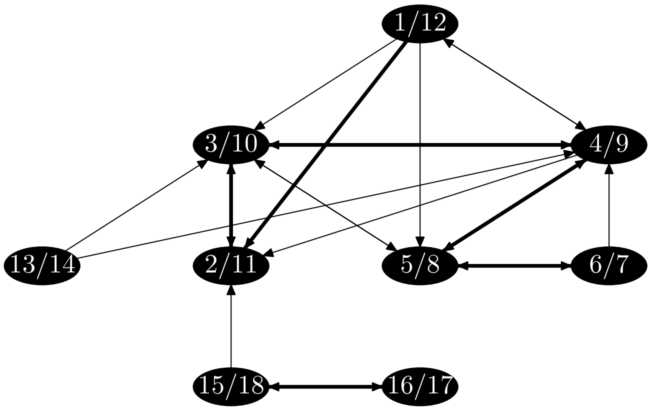
Lösungsvorschlag:
Ja, es gibt einen solchen Algorithmus: Durchlaufe die Kanten im Eingabegraphen und terminiere mit der Antwort *zyklisch*, sobald die n-te Kante erreicht wird. Diese Antwort ist richtig, denn ein kreisfreier ungerichteter Graph ist ein Wald und jeder Baum mit k Knoten hat genau $k-1$ Kanten.
Falls der Algorithmus jetzt noch nicht terminiert hat, wende die Methode auf Aufgabe T23 an,um die Kreisfreiheit in $O(|V|+|E|)=O(n)$ zu prüfen.

T25
Finden Sie mit dem Algorithmus von Kosaraju die starken Zusammenhangskomponenten im LuFGTI-Besuchsgraph:



Die beiden Tiefensuchen ergeben folgendes:





Man sieht somit, daß die starken Komponenten gerade $\{Q, A\}$, $\{V\}$ und $\{D, J, P, O, S, H\}$ sind.

H19 (10 Punkte)
Finden Sie einen Graphen mit zehn Knoten, 25 Kanten und möglichst vielen starken Zusammenhangskomponenten.

Lösungsvorschlag:
Wir bilden über den Knoten $v_1, \ldots, v_{10}$ zunächst den Graphen, der alle Kanten (v_i, v_j) mit $i < j$ enthält. Dieser Graph ist offensichtlich ein DAG und hat deshalb zehn starke Zusammenhangskomponenten. Natürlich können wir von den 45 Kanten 20 beliebige löschen, ohne daß diese Eigenschaft verletzt wird.
Eine schlechtere Lösung wäre beispielsweise folgende: Eine gerichtete Clique aus fünf Knoten hat $5 \cdot (5-1) = 20$ Kanten. Wenn wir an einen ihrer Knoten einen gerichteten Pfad aus fünf weiteren Knoten anhängen, ergeben sich insgesamt sechs Komponenten.

H20 (10 Punkte)
Entwerfen Sie einen Algorithmus, der für gerichtete Graphen $G = (V, E)$ in $O(|V| + |E|)$ entscheidet, ob jeder Knoten aus V auf einem gerichteten Kreis liegt.

Lösungsvorschlag:
Zunächst können wir die starken Zusammenhangskomponenten in der gewünschten Zeit ermitteln. Genau die Knoten, die auf keinem Kreis liegen, haben keine Schleife und bilden jeweils eine starke Zusammenhangskomponente der Größe eins.

H21 (10 Punkte)
Jemand behauptet, der Algorithmus zum topologischen Sortieren aus der Vorlesung funktioniert insofern auch auf Graphen, die Zyklen erhalten, als daß er Sortierungen mit der folgenden Eigenschaft liefert:

> Wenn zwei Knoten $u, v \in V$ in verschiedenen starken Zusammenhangskomponenten liegen, dann werden sie korrekt geordnet.

Beweisen Sie diese Aussage und überlegen Sie sich ein Anwendungsbeispiel, oder widerlegen Sie sie und entwerfen Sie einen Ersatzalgorithmus.

Lösungsvorschlag:
Die Aussage ist falsch. Dies sieht man leicht, indem man zwei gerichtete Kreise der Länge drei durch eine gerichtete Kante verbindet und die Suche an einem ungünstigen Knoten im ersten Kreis beginnt und dann zuerst in den zweiten Kreis hineinläuft, bevor man den ersten abarbeitet.
Der Ersatzalgorithmus kann zum Beispiel wie folgt aussehen.

1. Schrumpfe die starken Zusammenhangskomponenten jeweils zu einzelnen Knoten zusammen.

2. Sortiere diesen Komponentengraphen topologisch.

3. Expandiere die Komponenten in der ermittelten Reihenfolge. Hierbei spielt die Reihenfolge innerhalb jeder Komponente keine Rolle.

Offensichtlich leistet dieses Verfahren das Gewünschte.