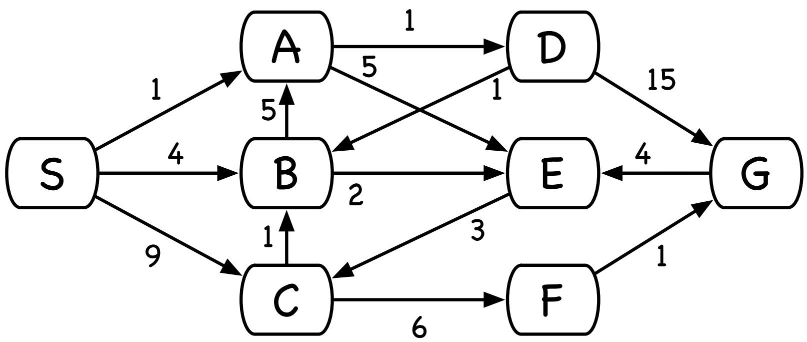
# AuD – Algorithmen und Datenstrukturen

## Aufgabenblatt 6 : Graphalgorithmen

Abgabe bis 4. Januar, 1600 Uhr – Besprechung am 6.-8. Januar 2016

| |
|---|
| |
| von |
| 10 |

**Übungsaufgabe 6.1:**




Betrachten Sie obigen Graphen $G$. Wenden Sie jeweils das verlangte Verfahren an bzw. beantworten Sie die Frage oder begründen Sie, warum dies nicht geht. (Bei den ersten sechs Teilaufgaben spielen die Kantengewichte keine Rolle.)

1. Ermitteln Sie mit der Breitensuche einen Breitensuchbaum. Starten Sie den Algorithmus bei $s$. (1 Pkt.)
2. Ist das Ergebnis der Breitensuche eindeutig? (1 Pkt.)
3. Ermitteln Sie mit der Tiefensuche einen Tiefensuchwald und insb. die Zeiten $d[v]$ und $f[v]$ für jeden Knoten $v$. Starten Sie den Algorithmus wieder bei $s$. (1 Pkt.)
4. Geben Sie eine topologische Sortierung des Graphen $G$ an. (1 Pkt.)
5. Bestimmen Sie mit dem Algorithmus aus der Vorlesung die starken Zusammenhangskomponenten von $G$. Geben Sie dazu das Ergebnis von $DFS(G)$ und $DFS(G^T)$ sowie die starken Zusammenhangskomponenten an. (1 Pkt.)
6. Geben Sie den Komponentengraphen von $G$ an. (1 Pkt.)
7. Interpretieren Sie $G$ nun als ungerichteten Graphen. Wenden Sie den Algorithmus von Kruskal an, um einen minimalen Spannbaum zu bestimmen. (Es genügt hier anzugeben welche Kanten in den minimalen Spannbaum aufgenommen werden und in welcher Reihenfolge dies geschieht!) (1 Pkt.)
8. Bestimmen Sie mit dem Algorithmus von Dijkstra kürzeste Pfade von $s$ zu allen anderen Knoten. Geben Sie hierzu jeweils den Graph mit den Werten von $d[v]$ nach jeder Iteration der while-Schleife an und machen Sie $\pi[v]$ z.B. durch dickere Kanten deutlich. (2 Pkt.)
9. Betrachten Sie den Sourcecode von Dijkstras Algorithmus und vom Algorithmus von Prim. Was ist der hauptsächliche Unterschied der beiden Algorithmen bei der Ausführung? (1 Pkt.)

| |
|---|
| von |
| 6 |

**Übungsaufgabe 6.2:**

1. Sei $G = (V, E)$ ein ungerichteter Graph. Bei einer $k$-Färbung von $G$ wird jedem Knoten $v \in V$ eine Farbe $c \in [k] = \{1, 2, \ldots, k\}$ zugewiesen. Die Färbung ist korrekt, wenn zwei benachbarte Knoten nie die gleiche Farbe haben. Formal ist eine *$k$-Färbung* dann also eine Abbildung $f : V \to [k]$ und eine *korrekte $k$-Färbung* $f$ erfüllt $f(x) \neq f(y)$ für jede Kante $\{x, y\} \in E$. Wir definieren folgendes Problem:

   $$k\text{-col} = \{\langle G\rangle \mid G \text{ besitzt eine korrekte } k\text{-Färbung}\}$$

   Angenommen Sie wissen, dass 3-col $NP$-vollständig ist. Zeigen Sie, dass auch 8-col $NP$-vollständig ist. (2 Pkt.)

2. Wir betrachten das folgende Spiel: Auf einem Feld mit $n$ Spalten, $n$ Zeilen und $n \times n$ Feldern können auf jeder der $n^2$ Positionen entweder ein weißer oder ein schwarzer Stein liegen oder das Feld kann leer sein. Eine Anfangsmarkierung besteht aus einer Zuweisung von Spielsteinen zu den Feldern. Das Spiel wird dann (alleine) gespielt, indem beliebig viele Steine entfernt werden. Ziel ist es, dass jede Spalte nur noch Steine einer Farbe enthält (oder leer ist) und jede Zeile mindestens einen Stein enthält. Ob man dieses Ziel erreichen kann (und damit das Spiel gewinnt), hängt von der Anfangsmarkierung ab. Die Eingabe ist ein Spiel $S$ bestehend aus dem Feld und der Anfangsmarkierung (für das Feld reicht die Zahl $n$). Die Eingabe wird akzeptiert, wenn $S$ gewonnen werden kann. Zeigen Sie, dass dieses Problem $NP$-vollständig ist. (4 Pkt.)

Informationen und Unterlagen zur Veranstaltung unter:

`http://www.informatik.uni-hamburg.de/TGI/lehre/vl/WS1516/AuD`