

---

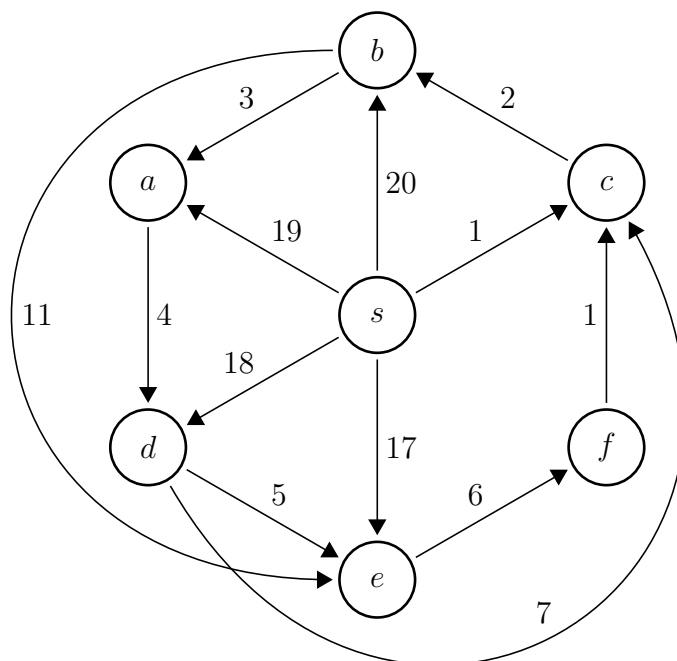
## Grundlagen: Algorithmen und Datenstrukturen

---

*Abgabetermin: Keine Abgabe und Korrektur*

### Tutoraufgabe 1

Führen Sie den Algorithmus von Dijkstra auf dem folgenden Graphen durch, um jeweils einen kürzesten Weg von  $s$  zu jedem anderen Knoten zu finden. Protokollieren Sie nachvollziehbar Ihre Vorgehensweise, und markieren Sie zum Schluss alle Kanten, die zum gefundenen Kürzeste-Wege-Baum gehören.



### Tutoraufgabe 2

Argumentieren Sie, warum es unklug ist, bei der Berechnung eines kürzesten Weges von  $s$  nach  $t$  in einem Graphen mit echt positiven Gewichten alle möglichen (einfachen) Wege aufzuzählen und deren Distanz zu berechnen.

Benutzen Sie dazu exemplarisch ein Gitter der Größe  $n$ . Die  $n^2$  Knoten werden durch die Tupel  $(x, y)$  mit  $x, y \in [n]$  identifiziert.

Die Kanten  $((x, y), (x + 1, y))$  seien gerichtet während die Kanten  $((x, y), (x, y + 1))$  bidirektional verlaufen. Berechnen Sie die Anzahl aller Pfade von  $s = (1, 1)$  zu  $t = (n, n)$ .

### Tutoraufgabe 3

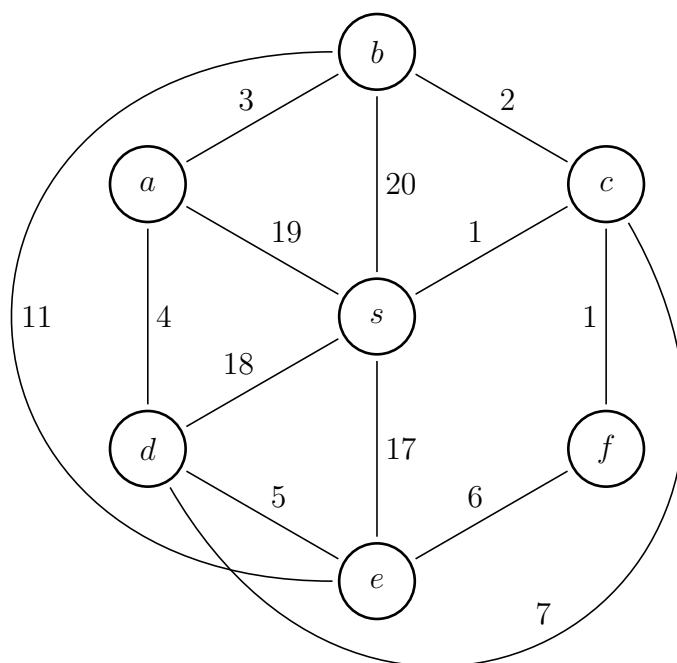
Geben Sie einen möglichst effizienten Algorithmus an, der die *Anzahl* der kürzesten Wege von einem Knoten  $s$  zu allen anderen Knoten in einem ungerichteten Graphen  $G$  mit echt positiven Gewichten berechnet.

#### Anmerkung zu den Hausaufgaben auf diesem Blatt:

Die Hausaufgaben dieses Blattes werden nicht korrigiert. Die Musterlösung wird am Freitag, dem 19. Juli, veröffentlicht.

#### Hausaufgabe 1

- (a) Führen Sie den Algorithmus von Prim mit Startknoten  $s$  auf dem folgenden Graphen durch, um einen minimalen Spannbaum zu finden. Protokollieren Sie nachvollziehbar Ihre Vorgehensweise, und markieren Sie zum Schluss alle Kanten, die zum gefundenen minimalen Spannbaum gehören.



*Anmerkung:* Der Algorithmus von Dijkstra und der Algorithmus von Prim sind, obwohl sie verschiedene Probleme lösen, vom Wesen her sehr ähnlich. Der einzige *wesentliche* Unterschied besteht in dem, was die Prioritäten der Knoten in der Priority-Queue aussagen, sowie in der Regel, nach der die Prioritäten aktualisiert werden. Bei Dijkstra entspricht die Priorität eines Knotens  $v$  der Länge eines bis zum aktuellen Zeitpunkt kürzesten gefundenen Pfades vom Startknoten zu  $v$ , während es bei Prim das kleinste Kantengewicht einer Kante aus der Menge der bereits fertig bearbeiteten Knoten in die Menge der noch unbearbeiteten Knoten ist.

- (b) Führen Sie den Algorithmus von Kruskal auf demselben Graphen wie in (a) durch, um einen minimalen Spannbaum zu finden. Protokollieren Sie nachvollziehbar Ihre Vorgehensweise, und markieren Sie zum Schluss alle Kanten, die zum gefundenen minimalen Spannbaum gehören.