

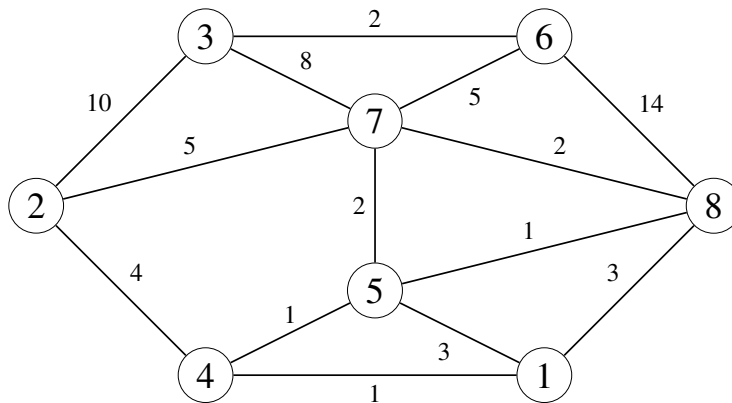


**Hausaufgaben zur Vorlesung »Algorithmen und Datenstrukturen«**

**Blatt 12**

**Hausaufgabe 12.1** (Dijkstra (3 Punkte))

Wenden Sie den Algorithmus von Dijkstra auf den nachfolgenden Graphen an um die kürzesten Wege von 1 zu jedem anderen Knoten zu bestimmen. Stellen Sie dabei die Veränderung der Werte  $v[j]$  tabellarisch dar und geben Sie an, welcher Knoten in dem entsprechenden Schritt gewählt wird. Zeichnen Sie anschließend den Baum der kürzesten Wege.



**Hausaufgabe 12.2** (Bellman-Ford (3 Punkte))

Sei  $G = (V, E)$  ein Graph. Zeigen Sie: Ist die minimale Zyklus-Länge echt größer Null und jeder Knoten von 1 aus erreichbar, so haben die Bellman-Gleichungen eine eindeutige Lösung.

**Hausaufgabe 12.3** (Eulertour (4 Punkte))

Zeigen Sie:

- Ein zusammenhängender ungerichteter Graph  $G = (V, E)$  enthält eine Eulertour genau dann, wenn jeder Knoten in  $V$  geraden Knotengrad hat.
- Ein stark zusammenhängender gerichteter Graph  $G = (V, E)$  enthält eine Eulertour genau dann, wenn für jeden Knoten  $v \in V$  gilt, dass  $d_{in}(v) = d_{out}(v)$ .

### **Programmieraufgabe 12.4**

Implementieren Sie die angegebenen Verfahren, um in ungerichteten Graphen kürzeste Wege von einem Startknoten zu einem Zielknoten zu finden.

a) Ohne Kantengewichte: Breitensuche

b) Mit Kantengewichten: Algorithmus von Dijkstra

Hinweise:

- Das Interface `'java.util.Queue'` wird z.B. von `'java.util.LinkedList'` unterstützt.
- Sie dürfen den Algorithmus von Dijkstra in der einfachen Variante umsetzen, in der die Menge  $Q$  als Liste oder Array abgebildet wird, können natürlich aber auch einen Heap (z.B. mittels `'java.util.PriorityQueue'`) verwenden.
- Wie in Programmieraufgabe 4 gibt jede Teilaufgabe 1/2 Punkt.

**Abgabe** der theoretischen Aufgaben Donnerstag, den 07. Juli, bis spätestens 14 Uhr. Die Abgabe der theoretischen Aufgaben erfolgt im Schrein.