

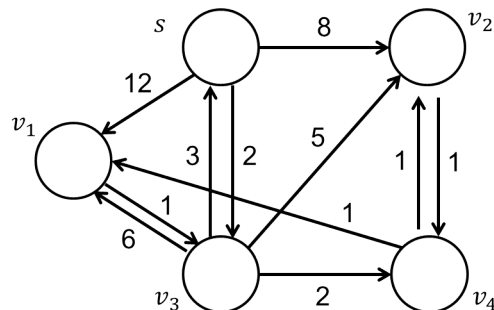
Übungen zur Vorlesung  
**Algorithmen und Datenstrukturen**  
WiSe 2019/20  
Blatt 12

Wichtige Hinweise:

- > Falls Sie bei der Bearbeitung einer Aufgabe größere Schwierigkeiten hatten und deswegen die Bearbeitung abgebrochen haben, so versuchen Sie bitte Ihre Schwierigkeiten in Form von Fragen festzuhalten. Bringen Sie Ihre Fragen einfach zur Vorlesung oder zur Übung mit!
- > Kursraum: <https://elearning.uni-regensburg.de/course/view.php?id=9228>

**Aufgabe 1:**

Führen Sie Dijkstras Algorithmus an folgendem Beispiel durch. Starten Sie mit dem Knoten  $s$  und geben Sie alle Zwischenschritte an. Demonstrieren Sie, wie sich der Min-Heap und das Vorgängerfeld Schritt-für-Schritt verändern.



Demonstrieren Sie anschließend an dem gleichen Beispiel den Algorithmus von Bellmann-Ford. Geben Sie an, in welcher Reihenfolge Sie die Kanten durchlaufen haben.

**Aufgabe 2:**

Viele Algorithmen (z.B. Prim's Algorithmus, Dijkstra's Algorithmus) benötigen eine Prioritätswarteschlange als Datenstruktur. Implementieren Sie eine Prioritätswarteschlange mit folgenden Operationen in C, C++, C# oder Java:

1. `Insert(elem e, key x)`: Fügt ein Element  $e$  mit Schlüssel  $x$  ein
2. `GetMin()`: Liefert ein Element mit kleinstem Schlüssel
3. `ExtractMin()`: Liefert und entfernt ein Element mit kleinstem Schlüssel
4. `DecreaseKey(elem e, pos x, key k)`: Verringert (!) den Schlüssel eines Elements  $e$  an Position  $x$  auf den Wert  $k$

Bestimmen Sie Zeit- und Platzkomplexität Ihrer implementierten Operationen!

**Aufgabe 3:**

Sei  $G = (V, E)$  ein beliebiger gerichteter Graph mit Gewichtung  $w : E \rightarrow \mathbb{R}$ . Warum kann der Algorithmus von Dijkstra nach dieser Definition nicht zur Berechnung kürzester Wege verwendet werden? Sei nun  $w^* = \min_{(u,v) \in E} \{w(u, v)\}$  das niedrigste Gewicht in  $G$  und  $\hat{w}(u, v) = w(u, v) - w^*$ . Zeigen oder widerlegen Sie, dass eine Anwendung des Algorithmus von Dijkstra in  $G = (V, E)$  mit Gewichtung  $\hat{w}$  und geeigneter Umrechnung kürzeste Wege in  $G = (V, E)$  mit Gewichtung  $w$  liefert.

**Aufgabe 4:**

Führen Sie den Algorithmus APSP für den folgenden gerichteten und gewichteten Graphen durch. Geben Sie jeweils nach jeder Iteration die Matrix  $L^{(i)}$  an.

