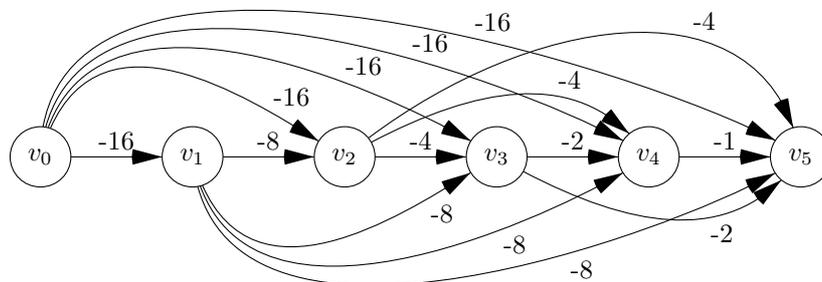


Ablauf der Übungen: Die Übungen sind weitgehend in die Vorlesung integriert. Es werden nicht alle Aufgaben besprochen und zum Teil nur Lösungshinweise gegeben. Sie haben jedoch die Möglichkeit, Abgaben zu machen, die dann auch korrigiert werden. Bei Interesse kann die Übung auch in den Vorlesungsblock Freitag 8h00-9h30 (Raum 0.124) ausgedehnt werden.

Eine allgemeine Empfehlung: Programmieren Sie die hier vorgestellten Algorithmen aus oder rechnen Sie zumindest unbedingt einige Beispiele durch, um die Eigenheiten der einzelnen Verfahren besser kennenzulernen.

1. (leicht) **Worst-Case-Beispiele für den Bellman-Ford-Algorithmus – Teil 1:** In folgendem Graphen sollen mit dem Bellman-Ford-Algorithmus kürzeste Wege vom Startknoten v_0 aus berechnet werden. Wieviel Schritte benötigt der Algorithmus? Wie hängt die Laufzeit von der Reihenfolge der Kanten in der Adjazenzliste ab (also davon, in welcher Reihenfolge adjazente Knoten in die FIFO-Datenstruktur eingefügt werden)?



2. (mittel–schwer) **Worst-Case-Beispiele für den Bellman-Ford-Algorithmus – Teil 2:** In obigem Beispiel hat der Graph $m \in \Theta(n^2)$ Kanten. Der Worst-Case kann aber auch für beliebige n und m erreicht werden. Versuchen Sie, zu gegebenen n und m Graphen mit n Knoten und m Kanten zu konstruieren, so dass die Kürzeste-Wege-Suche mit dem Bellman-Ford-Algorithmus $\Theta(n \cdot m)$ Schritte benötigt. (Literatur-Hinweis: Wenn Sie auf keine Lösung kommen, betrachten Sie das Beispiel in Mehlhorn, Näher: The LEDA Platform ..., <http://www.mpi-sb.mpg.de/~mehlhorn/LEDAbook.html>, Abschnitt 7.5.8, S.62ff)
3. (mittel–schwer) **Varianten des Bellman-Ford-Algorithmus:** Bei der Vorgängerheuristik vermeidet man das Bearbeiten von Knoten, für die man bereits weiß, dass sie im Folgenden nochmals in die FIFO aufgenommen und so zu einem späteren Zeitpunkt noch bearbeitet werden. Diese Idee lässt sich erweitern: Wird ein Wert $D(v)$ für einen Knoten v verringert, so werden alle Knoten im Unterbaum von v innerhalb des (bis dahin berechneten) Kürzeste-Wege-Baums später nochmal in der FIFO auftauchen, d.h., man sollte all diese Knoten aus der FIFO löschen, um unnötige Arbeit zu vermeiden. Überlegen Sie, wie man diese Idee umsetzen kann. Wird dadurch der Worst-Case verbessert?
4. (mittel) **Andere Varianten des Bellman-Ford-Algorithmus:** Eine weitere Idee zur Vermeidung unnötiger Arbeit geht auf D'Esopo und Pape zurück: Da man (wie in Aufgabe 3 beschrieben) weiß, dass bei Verringerung eines $D(v)$ die Knoten im Unterbaum alle nochmals bearbeitet werden müssen, werden diese bevorzugt behandelt. Wir erweitern dazu die FIFO zu einer beidseitigen Schlange: Hatte ein Knoten $D(v)$ vor der Verringerung schon einen endlichen Wert, so wird der Knoten vorne in die Schlange eingefügt, ansonsten am Ende. Der nächste Knoten wird stets vorne entnommen, so dass zunächst die bisher falsch berechneten Teile des Kürzeste-Wege-Baums korrigiert werden, bevor neue Knoten erstmals betrachtet werden. Spielen Sie diesen Algorithmus für einige typische Beispiele und den Graphen von Aufgabe 1 durch.

Überlegen Sie auch hier, welchen Einfluss die Reihenfolge der Kanten in der Adjazenzliste des Graphen hat.