Algorithmen und
Berechnungskomplexität I
Wintersemester 2022/23

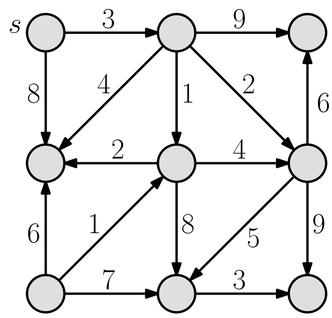
Prof. Dr. Anne Driemel
Frederik Brüning, Jan Eube
Institut für Informatik

Abgabe: 21.12.2022 bis 10.00 Uhr

Übungsblatt 10

Aufgabe 10.1: Algorithmus von Dijkstra (3+3=6 Punkte)

(a) Bestimmen Sie mithilfe des Algorithmus von Dijkstra für den folgenden Graphen die Werte $\delta(s,v)$ für alle Knoten v und geben Sie einen Kürzeste-Wege-Baum mit Wurzel s an.



(b) Zeigen Sie, dass der Algorithmus von Dijkstra die Abstände $\delta(s,v)$ für Graphen mit negativen Kantengewichten im Allgemeinen nicht korrekt berechnet, auch wenn diese keinen negativen Kreis enthalten.

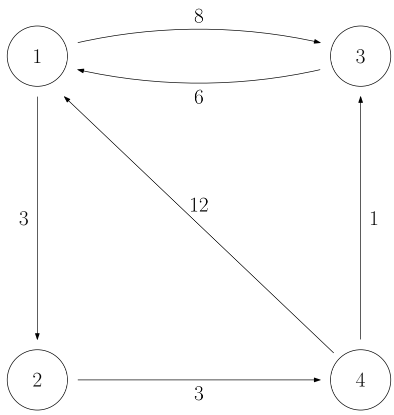
Aufgabe 10.2: Eigenschaften des Algorithmus von Dijkstra (4 Punkte)

Sei $G = (V, E)$ ein gerichteter Graph mit nichtnegativer Kantengewichtung w. Wir führen den Algorithmus von Dijkstra für einen Knoten $s \in V$ aus. Ohne Einschränkung sei jeder Knoten des Graphen von s aus erreichbar. Mit v_i bezeichnen wir den Knoten von G, der als i-ter Knoten zur Menge S hinzugefügt wird. Sei d_i der Wert von $d(v_i)$ zum Zeitpunkt des Hinzufügens von v_i zu S. Zeigen Sie induktiv, dass $d_i \leq d_{i+1}$ für alle $i = 1, \ldots, n-1$ gilt.

Aufgabe 10.3: Algorithmus von Floyd und Warshall (5 Punkte)

Wenden Sie auf dem nachfolgend abgebildeten gerichteten Graphen den Algorithmus von Floyd und Warshall an, um das All-Pairs-Shortest-Path-Problem zu lösen. Geben Sie dabei alle Zwischenergebnisse an, d.h. erstellen Sie für alle Werte von k eine Tabelle, aus der die aktuellen Distanzwerte und Vorgängerknoten hervorgehen.

Hinweis: Machen Sie sich klar, dass sich nach der Initialisierung in jeder folgenden Iteration maximal 6 Tabellenzellen verändern können!



Aufgabe 10.4: Topologische Sortierung (3 + 1 + 1 = 5 Punkte)

Eine Funktion $t\colon V \to \{1, \ldots, |V|\}$ auf den Knoten eines gerichteten Graphen $G = (V, E)$ heißt *topologische Sortierung*, falls $t(v) < t(w)$ für alle Kanten (v, w) von G gilt. Veranschaulicht bedeutet das, dass alle Kanten von links nach rechts verlaufen, wenn wir jeden Knoten v an Position $t(v)$ auf der Zahlengerade eintragen.

(a) Betrachten Sie den folgenden Algorithmus zur Ermittlung einer topologischen Sortierung.

```
Topsort (G, t)
Setze count = 0.
while (G ist nicht leer) do
    Finde einen Knoten v in G mit Ingrad 0.
    if (v = null) then return false
    Erhöhe count um 1.
    Setze t(v) = count.
    Entferne v und alle Kanten der Gestalt (v, w) aus G.
end while
return true
```

Beweisen Sie, dass Topsort eine *topologische Sortierung* von G erzeugt, falls G keinen Kreis besitzt.

Hinweis: Zeigen Sie zunächst, dass jeder azyklische Graph einen Knoten mit Ingrad 0 besitzt. Dazu können Sie die Kontraposition dieser Aussage zeigen: Wenn der Ingrad jedes Knotens mindestens 1 ist, dann besitzt der Graph einen Kreis. Überlegen Sie sich einen konstruktiven Beweis dieser Aussage.

(b) Beweisen Sie, dass G genau dann eine topologische Sortierung besitzt, wenn G keinen Kreis enthält.

(c) Beschreiben Sie, wie man Topsort in $O(|V| + |E|)$ Zeit implementieren kann.