



Übungsblatt 10

Abgabe via Moodle.
Deadline Fr. 14ter July

Aufgabe 1 (Tiefensuche iterativ, 6 Punkte)

Entwerfen Sie eine nicht rekursive Tiefensuche ausgehend von einem Knoten s und geben Sie **Pseudocode** an! Die Laufzeit $O(m + n)$ darf nicht überschritten werden.

Aufgabe 2 (Finden von Kreisen in Graphen, 2 + 2 + 2 Punkte)

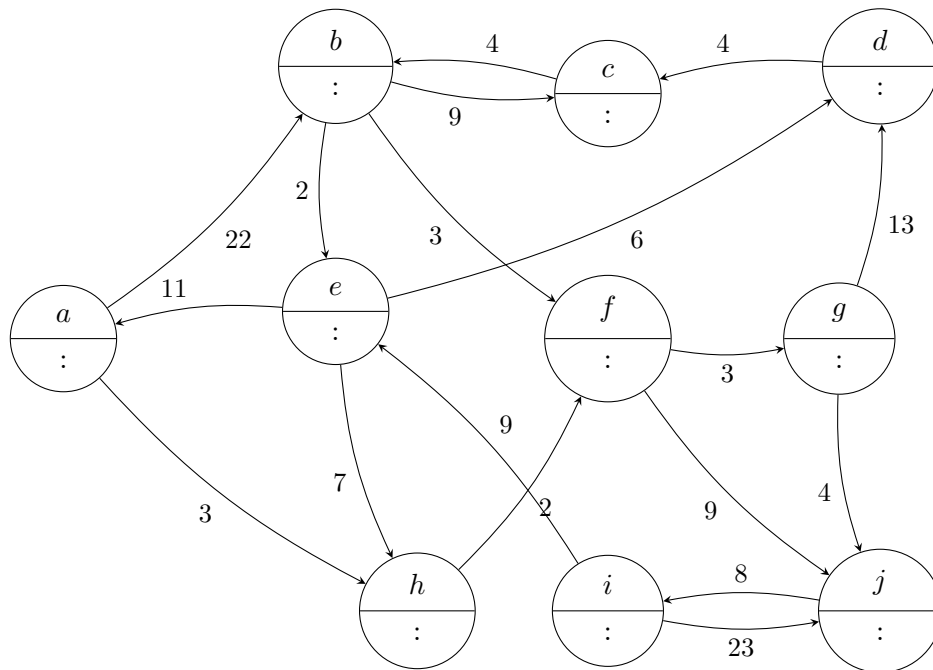
Mittels einer modifizierten Tiefensuche kann man in $O(n)$ Zeit bestimmen, ob ein **ungerichteter** Graph mit n Knoten einen Kreis enthält oder nicht. Gehen Sie davon aus, dass der Graph als Adjazenzfeld mit bigerichteten Kanten dargestellt wird.

1. Geben Sie an, wie Sie die Tiefensuche modifizieren würden, um einen entsprechenden Algorithmus zu erhalten.
2. Zeigen Sie, dass Ihr Algorithmus tatsächlich das gewünschte Laufzeitverhalten aufweist.
3. Liefert Ihr Algorithmus auch für **gerichtete** Graphen ein korrektes Ergebnis? Wenn nein, wie muss der Algorithmus modifiziert werden? Welches asymptotische Laufzeitverhalten hat der (möglicherweise modifizierte Algorithmus) für **gerichteten** Graphen schlimmstenfalls? Begründen Sie jeweils kurz.

Aufgabe 3 (Dijkstras Algorithmus, 4 Punkte)

Führen Sie auf folgendem Graphen den Algorithmus von Dijkstra aus, beginnend mit Knoten a . Als Ergebnis soll in jedem Knoten folgendes stehen: links vom Doppelpunkt die Nummer des Schritts, in dem der Knoten gescannt wurde; rechts vom Doppelpunkt die Länge des kürzesten Wegs von a . Zeichnen Sie außerdem den Baum der kürzesten Wege von a aus ein.

Sie können direkt in dieses Blatt einzeichnen.



Aufgabe 4 (*Kürzeste Pfade*, 2 + 2 + 2 + 2 Punkte)

1. Geben Sie einen gewichteten gerichteten Graph mit positiven Kantengewichten an, der einen Knoten v^* enthält, dessen vorläufige Distanz von Dijkstras Algorithmus mindestens dreimal mittels *decreaseKey* verringert wird. Geben Sie zusätzlich zum Graph auch den Knoten v^* und den Startknoten einer entsprechenden Dijkstra-Suche an.
2. Zeigen Sie: In gewichteten gerichteten Graphen sind Teilpfade von kürzesten Pfaden wiederum kürzeste Pfade.

Aufgabe P10 (*Dijkstra: Shortest Reach, optional*)

Für die praktischen Übungen verwenden wir die Plattform www.hackerrank.com. Hier müssen Sie sich registrieren um an den Übungen teilzunehmen. Unter dem Link

<https://www.hackerrank.com/adsi-2023>

finden die praktischen Übungen in der Form eines Programmierwettbewerbs statt.

In der zehnten Challenge sind Sie gefragt den Algorithmus von Dijkstra zu implementieren.

Auf Moodle finden Sie wieder ein Framework für diese Aufgabe *framework10.cpp*, welches Sie statt des vorgegebenen Codes verwenden können.

Genauere Informationen, sowie ein Beispiel finden Sie auf HackerRank.