Lehrstuhl für Informatik I
Algorithmen & Komplexität
Universität Würzburg

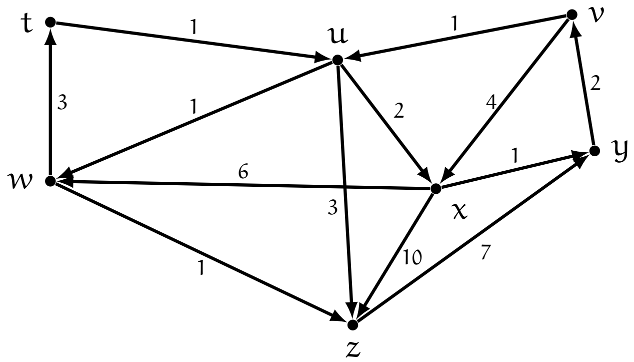
Würzburg, den 29. April 2020

Prof. Dr. Alexander Wolff
Diana Sieper, B. Sc.

2. Übungsblatt zur Vorlesung Algorithmische Graphentheorie (Sommer 2020)

Aufgabe 1 – Algorithmus von Dijkstra

a) Ermitteln Sie für den abgebildeten Graphen mit Hilfe des Algorithmus von Dijkstra einen Baum kürzester Wege bezüglich x. Geben Sie in nachvollziehbarer Art und Weise die Zwischenschritte an. **4 Punkte**



b) Geben Sie einen Graphen an, der gerichtete Kanten negativer Länge enthält, und bei dem der Algorithmus von Dijkstra versagt.

Konstruieren Sie ferner einen Graphen $G = (V, E)$, der gerichtete Kanten negativer Länge enthält und einen Knoten s, von dem aus jeder Knoten $v \in V$ erreichbar ist, bei dem der Algorithmus von Dijkstra mit Startknoten s aber ein korrektes Ergebnis liefert. G soll $n \geq 3$ Knoten und $m \geq n + 3$ Kanten enthalten.

Erklären Sie Ihre Lösungen kurz. **4 Punkte**

Aufgabe 2 – Kreissuche

Gegeben sei ein zusammenhängender ungerichteter Graph $G = (V, E)$. Wir interessieren uns hier für *einfache Kreise*, d. h. Kreise, die jede Kante höchstens einmal enthalten, mit einer Länge von mindestens 3.

Modifizieren Sie die Breitensuche so, dass der resultierende Algorithmus ermittelt, ob ein zusammenhängender Graph einen einfachen Kreis mit einer Länge von mindestens 3 enthält. Der modifizierte Algorithmus soll eine Laufzeit von $\mathcal{O}(|V|)$ aufweisen – im Gegensatz zur Breitensuche, die $\mathcal{O}(|V| + |E|)$ Zeit benötigt.

a) Geben Sie den Algorithmus in Worten und in Pseudocode an. **4 Punkte**

b) Zeigen Sie, dass Ihr Algorithmus die Anforderungen (d. h. Korrektheit und Laufzeit) erfüllt. **4 Zusatzpunkte**

c) Erklären Sie, wie sich der Algorithmus auf nicht notwendigerweise zusammenhängende ungerichtete Graphen verallgemeinern lässt. **1 Zusatzpunkt**

Aufgabe 3 – Metrisches TSP

In der Vorlesung wurde ein Approximationsalgorithmus auf Basis von Spannbäumen für das metrische TSP vorgestellt. Dieser besitzt *Güte* 2, d.h. die Rundreise, die der Algorithmus liefert, ist höchstens doppelt so lang wie eine optimale Rundreise.

Im Folgenden sei G ein vollständiger, metrischer, ungerichteter Graph mit Kantengewichten $c\colon E \to \mathbb{R}_{\geq 0}$. Betrachten Sie den folgenden Algorithmus:

`CompleteHamilton`(Graph $G = (V, E)$, Kostenfunktion c)
  wähle $s \in V$ beliebig
  wähle eine Kante $\{s, t\} \in E$ mit $c(s, t)$ minimal
  $C \leftarrow \langle\{s, t\}, \{t, s\}\rangle$
  while $V(C) \neq V$ **do**
    wähle einen Knoten $v \in V \setminus V(C)$, dessen Abstand zu C am geringsten ist
    sei w ein Knoten in C mit $c(v, w)$ minimal
    sei u einer der beiden Nachbarn von w auf C
    ersetze in C die Kante $\{u, w\}$ durch die Kantenfolge $\langle\{u, v\}, \{v, w\}\rangle$
  return C

a) Zeigen Sie, dass der Algorithmus von Prim (beginnend mit Startknoten s) in jedem Schritt denselben Knoten zum aktuellen Baum hinzufügt wie Algorithmus `CompleteHamilton` zum Kreis C. **3 Punkte**

   Hinweis: Sie dürfen vernachlässigen, dass es in einem Schritt mehrere gleich gute Wahlmöglichkeiten geben kann.

b) Zeigen Sie, dass Algorithmus `CompleteHamilton` Güte 2 besitzt. **5 Punkte**

   Hinweis: Verwenden Sie Teilaufgabe a) sowie die *Dreiecksungleichung*.

Bitte laden Sie Ihre Lösungen als pdf bis **Dienstag, 05. Mai 2020, 12:00 Uhr** im WueCampus-Kursraum beim entsprechenden Übungsblatt hoch. Geben Sie stets die Namen aller (bis zu 2) BearbeiterInnen an.

Begründen Sie stets Ihre Behauptungen und kommentieren Sie Ihren Pseudocode!