Lehrstuhl II für Mathematik
Prof. Dr. Arie M.C.A. Koster
Nils Spiekermann, M.Sc.

WS 2017/18

OPTIMIERUNG B
Programmieraufgaben 1
Abgabe bis 20.11.2017

Alle Beispielgraphen werdem im Lernraum im "LEMON Graph Format"[1] zur Verfügung gestellt. Es werden ausschließlich (bei der Korrektur) lauffähige und ausreichend dokumentierte Programme in C++ oder Python bewertet. Zudem ist es nicht gestattet Bibliotheken oder Pakete zu verwenden, die die Aufgaben untergraben. Bitte senden Sie ihre Ergebnisse unter spiekermann@math2.rwth-aachen.de ein.

Aufgabe 1: **2 Punkte**
Implementieren Sie einen effizienten Algorithmus, welcher einen minimalen Spannbaum eines gegebenen Graphen mit Kantengewichten berechnet. Geben Sie die Laufzeit des Algorithmus an.

Aufgabe 2: **2 Punkte**
Implementieren Sie einen effizienten Algorithmus, welcher einen kürzesten s-t-Weg in einem gegebenen gerichteten Graphen mit nicht-negativen Bogengewichten berechnet. Geben Sie die Laufzeit des Algorithmus an.

Aufgabe 3: **4+2+1 Punkte**
Sei $G = (V, E)$ ein Graph mit nicht-negativen Kantengewichten und $T \subset V$ eine Teilmenge der Knoten. Wir suchen einen minimalen Teilgraphen von G, welcher alle Knoten von T verbindet (dieser kann Knoten aus $V \backslash T$ enthalten). Für $T = V$ ist dieses Problem identisch zum minimalen Spannbaum Problem und für $T = \{s, t\}$ zum kürzesten s-t-Weg Problem.

a) Entwickeln Sie einen effizienten Algorithmus welcher eine gute Lösung findet (also eine Heuristik) und geben Sie seine Laufzeit an. Es ist unbekannt ob ein effizienter Algorithmus existiert, welcher eine optimale Lösung garantiert.
Tipp: Verwenden Sie Ihre Ergebnisse aus den vorherigen Aufgaben.

b) Erklären Sie die Idee Ihres Algorithmus.

c) Geben Sie einen Graphen an, auf welchem Ihr Algorithmus keine optimale Lösung findet.

Aufgabe 4: **3+1+5 Punkte**

a) Bestimmen Sie für jeden im Lernraum zur Verfügung gestellten Graphen einen minimalen Spannbaum. Geben Sie dessen Kosten, sowie die benötigte Rechenzeit an.

b) Bestimmen Sie einen möglichst kleinen Teilgraph von "graph1" welcher alle Knoten in "graph1_terminals" verbindet.

c) Die Dateien "graph*_terminals" enthalten eine Teilmenge der Knoten von "graph*", welche mit einem minimalen Teilgraph verbunden werden sollen. Benutzen Sie den Algorithmus aus der vorherigen Aufgabe um eine obere Kostenschranke für einen solchen zu berechnen. Geben Sie zudem die benötigte Rechenzeit an.

[1] http://lemon.cs.elte.hu/pub/tutorial/a00018.html