Albert-Ludwigs-Universität
Institut für Informatik
Prof. Dr. F. Kuhn
M. Fuchs, G. Schmid

Algorithmen und Datenstrukturen
Sommersemester 2024
Übungsblatt 9

Abgabe: Dienstag, 25. Juni, 2024, 10:00 Uhr

Aufgabe 1: Eindeutige Minimale Spannbäume *(10 Punkte)*

Sei $G = (V, E, w)$ ein *ungerichteter, zusammenhängender, gewichteter* Graph mit paarweise veschiedenen Kantengewichten.

(a) Zeigen Sie, dass G einen *eindeutigen* minimalen Spannbaum hat. *(5 Punkte)*

(b) Zeigen Sie, dass man durch die folgende Konstruktion den minimalen Spannbaum T' von G erhält:

Starte mit $T' = \emptyset$. Für jeden Schnitt in G, füge die leichteste Schnittkante zu T' hinzu.

Hinweis: Eine Möglichkeit ist hier über einen MST T von G zu argumentieren. Genauer gesagt, zu zeigen ist dass wenn eine Kante $e \in T'$ ist, auch $e \in T$ sein muss wie auch umgekehrt wenn $e \in T$ ist, dann muss ebenso $e \in T'$. *(5 Punkte)*

Aufgabe 2: Problem des Handlungsreisenden *(10 + 5* Punkte)*

Seien $p_1, \dots, p_n \in \mathbb{R}^2$ Punkte in der euklidischen Ebene. Der Punkt p_i gibt die Position von Stadt i an. Die Distanz zwischen zwei Städten i und j entspricht der euklidischen Distanz der entsprechenden Punkte p_i, p_j. Eine *Rundreise* ist eine Reihenfolge von Städten, so dass keine Stadt ausser der ersten mehr als einmal besucht wird. Gesucht ist die Rundreise welche die zurückgelegte Distanz minimiert. Dieses Problem ist im Allgemeinen sehr schwer.[1] Wir geben uns deshalb mit einer Rundreise zufrieden die höchstens doppelt so lang ist wie die minimale Rundreise.

Wir können das auch als Graphenproblem mit $G = (V, E, w)$ auffassen, wobei $V = \{p_1, \dots, p_n\}$ und $w(p_i, p_j) := \|p_i - p_j\|_2$. Damit ist G *ungerichtet und vollständig* und es gilt die *Dreiecksungleichung.*[2] Gesucht ist eine Rundreise $(i_1, ..., i_n)$ mit kleiner Gewichtssumme $w(p_{i_n}, p_{i_1}) + \sum_{j=1}^{n-1} w(p_{i_j}, p_{i_{j+1}})$.

(a) Sei G ein Graph wie oben beschrieben. Zeigen Sie, dass die Knoten eines minimalen Spannbaumes in Pre-order Reihenfolge (ausgehend von einer beliebigen Wurzel) eine Rundreise in G ergibt die höchstens doppelt so lang ist wie die minimale Rundreise. *(5 Bonus Punkte)*

(b) Implementieren Sie einen Algorithmus der eine Pre-order Reihenfolge eines minimalen Spannbaumes von G berechnet. Sie dürfen dazu die Vorlagen `TSP.py` und `AdjacencyMatrix.py` sowie Module für

[1] Das (exakte) Problem des Handlungsreisenden ist aus der sogenannten Klasse der $\mathcal{NP}$-vollständigen Probleme, für deren Lösung wahrscheinlich kein Polynomialzeitalgorithmus existiert (nur falls $\mathcal{P} = \mathcal{NP}$ was noch niemand zeigen oder widerlegen konnte). Mehr dazu in der Vorlesung zu theoretischer Informatik.

[2] Die Dreiecksungleichung impliziert, dass die direkte Kante zwischen zwei Knoten $a, b \in V$ höchstens so lang ist wie ein Umweg über einen dritten Knoten $c \in V$, d.h. $w(a, b) \le w(a, c) + w(c, b)$.

Heap und Union-Find Datenstrukturen benutzen[3]. Lesen Sie den Beispielgraph aus `cities.txt` als Adjazenzmatrix ein und wenden Sie ihren Algorithmus darauf an. Berechnen und notieren Sie die Gewichtssumme der resultierenden Rundreise (s.o.) in Ihren `erfahrungen.txt`. *(10 Punkte)*

[3]Zum Beispiel `heapq` und `networkx.utils.union_find`. Bei `heapq` entspricht `heappush` der `insert` und `heappop` der `delete-min` Operation aus der Vorlesung. Dabei können `heappush` und `heappop` auf Python-Listen angewendet werden (mehr Details hier). Wenn Sie ein Objekt `uf` der Klasse `UnionFind` instantiiert haben, legt `uf[`i`]` eine neue Menge $\{i\}$ an falls i noch nicht in `uf` existiert und liefert sonst einen Repräsentanten der Menge zurück in der i enthalten ist (dies kombiniert also die Funktionen von `make-set` und `find`, mehr Details hier).

Blatt 9

7)

a) Durch den Prim Algorithmus lässt sich zu jedem ungerichteten, zusammenhängenden, gewichteten Graphen ein minimaler Spannbaum erstellen. Dafür wird in jedem Schritt dem Spannbaum die leichteste Kante hinzugefügt. Wenn die Kantengewichte alle unterschiedlich sind, so steht dann jeweils pro Schritt nur eine Kante zur Auswahl und es existiert nur ein eindeutiger Spannbaum.

b) Angenommen es existiert ein minimaler Spannbaum T und T' wird nach Vorraussetzung in jedem Schritt die leichteste Kante hinzugefügt, so besitzen T und T' die gleichen Kanten. Sollte T' andere Kanten beinhalten als T, so wäre $w(T) \geq w(T')$. Daraus würde folgen, dass T kein minimaler Spannbaum ist was im Widerspruch zur Annahme steht.