Prof. für Algorithmen
und Datenstrukturen
Prof. Dr. Hannah Bast
Dr. Patrick Brosi

Algorithmen und Datenstrukturen
SS 2025

https://ad-wiki.informatik.uni-freiburg.de/teaching

UNI FREIBURG

Übungsblatt 10

Abgabe bis Dienstag, den 15. Juli um 12:00 Uhr

Aufgabe 1 (10 Punkte)

Implementieren Sie die Funktion *shortest_path* in der auf dem Wiki verlinkten Datei *route_planner.py*, mittels Dijkstras Algorithmus, wie in der Vorlesung erklärt. Verwenden Sie dabei das in der Vorlesung geschriebene Modul *graph.py* und Ihre Implementierung einer *PriorityQueue* vom Ü9. Falls Sie mit dem Ü9 Probleme hatten oder es nicht gemacht haben, können Sie auch die Musterlösung verwenden. Es ist Ihnen überlassen, ob Sie *change_key* verwenden oder bei Bedarf denselben Knoten mehrfach einfügen, wie in der Vorlesung erklärt. Die Funktion *shortest_path* sollte eine geordnete Liste aller Knoten auf dem kürzesten Weg zurückgeben (Startknoten zuerst, Endknoten zuletzt).

Aufgabe 2 (10 Punkte)

Implementieren Sie die Funktion *main* in der Datei *route_planner.py*. Die Funktion soll den Graphen und die Adressen aus den per Kommandozeilenargument übergebenen Dateien einlesen. Nach dem Einlesen soll man wiederholt einen Start- und einen Endknoten eingeben können, mittels der Adressensuche vom Ü8. Falls Sie mit dem Ü8 Probleme hatten oder es nicht gemacht haben, können Sie auch die Musterlösung verwenden. Es soll dann mittels der *shortest_path* Funktion von Aufgabe 1 der kürzesten Weg zwischen diesen beiden Knoten berechnet werden. Weitere Details des Eingabemodus sind in der Codevorlage erklärt.

Der kürzeste Weg sollte als URL mit den Koordinaten der Knoten auf dem Weg ausgegeben werden. Das Format dieser URL ist in der Codevorlage erklärt. Falls der kürzeste Weg $n > 100$ Knoten hat, sollte die URL nur die Koordinaten jedes $\lceil n/100 \rceil$-ten Knotens auf dem Weg enthalten, damit die URL nicht zu lang wird. Die Koordinaten des Start- und Zielknotens sollten aber immer enthalten sein. Wenn Sie die URL in Ihren Browser eingeben, wird der Verlauf des kürzesten Weges auf einer Karte visualisiert. Das wird Ihnen dabei helfen, Fehler zu finden bzw. sich schließlich von der Korrektheit Ihres Codes zu überzeugen.

[kürzesten Weg zur Rückseite berechnen und dann zielstrebig folgen]

Auf dem Wiki sind drei Graphen verschiedener Größe verlinkt: für Freiburg Stadt, für den Regierungsbezirk Freiburg und für ganz Baden-Württemberg. Um die volle Punktzahl zu erreichen, reicht es, wenn Ihr Programm auf dem Graphen Freiburg Stadt korrekt funktioniert.

Committen Sie Ihren Code und Ihre *erfahrungen.txt* (mit den üblichen Informationen) in unser SVN, in einen neuen Unterordner *blatt-10*. Die Straßengraphen bitte nicht mit hochladen, das gibt sonst einen Punkt Abzug pro Megabyte.

Edsger Dijkstra war ein ziemlich spezieller Typ. Was finden Sie am speziellsten?