

Übungsblatt 9

Abgabe: 2.07.2007

Aufgabe 1 Routenplanung EWD

Um möglichst klimaschonend von Astadt nach Bdorf zu kommen, ist es nur logisch, den kürzesten Weg zwischen diesen Orten zu benutzen. Implementiert einen Routenplaner, der die günstigste Strecke zwischen zwei Orten mit Hilfe des Dijkstra-Algorithmus ermittelt. Dabei sollen Strukturen der Serie 6 (Transitionssystem) ggf. modifiziert wiederverwendet werden.

Die Knoten repräsentieren die Städtenamen. Ihr dürft die Klasse **Zustand** um die int-Variable **Entfernung** erweitern, die zeigt, wie weit es vom Startzustand zu diesem im Dijkstra-Algorithmus ist. Die Entfernung "-1" bedeutet, dass es keine Verbindung gibt, und alle Entfernungen ≥ 0 repräsentieren die tatsächliche Entfernung. Erweitert die Klasse mit **get()** und **set()** Methoden für private Variablen wenn notwendig. Ändert die **equals()** Methode entsprechend. Ändert die Methode **traversiere()** so, dass sie alle Transitionen als Vector zurückliefert.

Erstellt eine Klasse **Entfernung**, die **Markierung** erweitert.

Erweitert **Markierung** um einen Konstruktor, der einen String (die Entfernung) als Integer Object in der Variable **result** speichert. Ändert die Methode **equals()** entsprechend.

In der Klasse **Transition** erstellt public Methoden, die die Entfernung und den Zielzustand der Transition auslesen.

In der Klasse **TransitionsSystem** könnt ihr alle Variablen und Methoden auf die Sichtbarkeit **protected** stellen.

Die Klasse **EWDRoute** überschreibt die Methode **traversiereTransitionsSystem()**. Implementiert dort den Dijkstra-Algorithmus. Verwendet dort die Zustände und die geänderte Methode **traversiere()** der Zustände.

Zum Testen erstellt die Methode **generiereSystem()** in **EWDRoute**, die den Graphen aufbaut. Dieser Verbindungsgraph für einige deutsche Städte erhaltet ihr von der auf der PI-2-Webseite bereitgestellten Klasse *Verbindungsgraph*.

Als Eingabe des Startzustands sollt ihr die Methode **setzeAnfangsZustand()** verwenden, um mit dem Dijkstra-Algorithmus als Ergebnis den günstigsten Weg zum Zielzustand zu bekommen. Erstellt eine Methode, die als Parameter einen **String** erwartet, der den Namen der Ziel-Stadt darstellt. Diese soll anschliessend als Startzustand (und die jeweiligen Entfernungen der Städte entsprechend) gesetzt sein.