

Der Algorithmus von Dijkstra

Programm entsprechend der Vorlage von Gallenbacher¹

Algorithmus – Beschreibung bei Gallenbacher:

0. Markiere die Startstadt rot, weise ihr die Kennzahl 0 zu. Bezeichne diese als aktuelle Stadt.
1. Gehe aus von der aktuellen Stadt zu allen direkt erreichbaren Nachbarstädten.
... und führe das Folgende für jede Nachbarstadt durch:
 - Errechne die Summe aus der Kennzahl an der aktuellen Stadt und der Länge der Strecke dorthin.
 - Ist die Nachbarstadt bereits rot markiert, mache nichts.
 - Hat die Nachbarstadt keine Kennzahl, weise ihr die Summe als Kennzahl zu. markiere die Strecke zur aktuellen Stadt.
 - Hat die Nachbarstadt eine Kennzahl größer der Summe, streiche die dortige Kennzahl sowie die Markierung. Weise ihr danach die Summe als neue Kennzahl zu. Markiere die Strecke zur aktuellen Stadt.
2. Betrachte alle Städte, die zwar eine Kennzahl haben, aber noch nicht rot markiert sind. Suche die Stadt mit der kleinsten Kennzahl.
3. Bezeichne diese als aktuelle Stadt. Weisen mehrere Städte die kleinste Kennzahl auf, wähle eine beliebige davon als aktuelle Stadt.
4. Markiere die aktuelle Stadt rot, zeichne die dort markierte Strecke in rot ein.
5. Falls es noch Städte gibt, die nicht rot markiert sind, weiter bei (1.)

Diese Formulierung lässt sich nicht ganz einfach in Scheme umsetzen, da sie sich an prozeduraler / objektorientierter Programmierung orientiert und nicht an funktionaler. Insbesondere der Punkt 5 ist ärgerlich, hier sollte besser eine Formulierung wie "...bis es keine Städte mehr gibt, die nicht rot markiert sind" gewählt werden.

1 Jens Gallenbacher: Abenteuer Informatik; Spektrum Akademischer Verlag