

Algorithmen und Datenstrukturen

Sommersemester 2009

2. Übungsblatt

Aufgabe 1: Greedy-Strategie

In einem schleswig-holsteiner Landkreis bestehend aus einer Ansammlung von Städten und Dörfern (als Punkte in Figure 1) sollen neue Schulen entstehen, damit Schüler nicht mehr in andere Landkreise pendeln müssen. Bei der Bauplanung ist entscheidend, dass kein Schüler mehr als 20km zu seiner neuen Schule reisen muss. Es gibt demnach zwei grundlegende Anforderungen:

- a) Jede Schule muss in einer Stadt oder einem Dorf sein.
- b) Die maximale Entfernung von Städten und Dörfern ohne Schule zur nächsten Schule ist mit 20km begrenzt. Städtepaare, die dieser Anforderung genügen, sind im rechten Teil von Figure 1 verbunden.

Die Bauplanung ist nun mit der Frage beschäftigt, was die minimale Anzahl der benötigten Schulen ist.

- a) Geben Sie einen Greedy-Algorithmus im Pseudocode an, der den Landkreis mit möglichst wenig Schulen versorgt. Hinweis: Geben Sie Pseudocode an und kein komplexes Java-Programm! (4 Punkte)
- b) Bestimmen Sie die Laufzeit des Algorithmus. (2 Punkte)
- c) Ist ihr Algorithmus optimal? Beweisen bzw. widerlegen Sie die Optimalität. (2 Punkte)

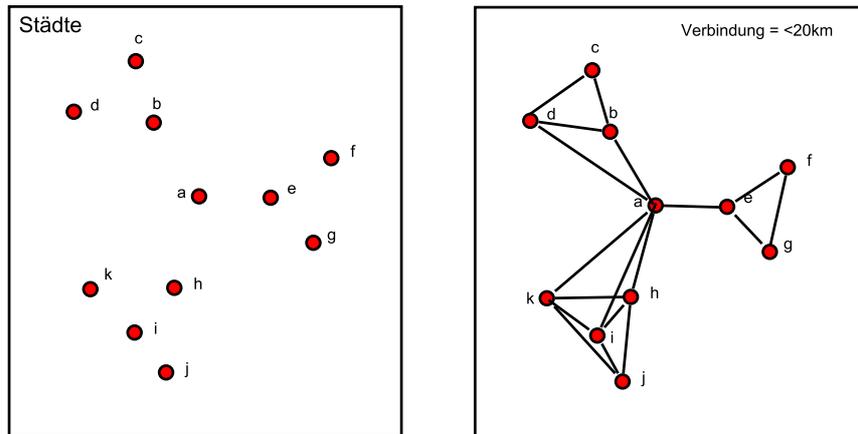


Abbildung 1: Skizze der Lage der Städte und Dörfer

Aufgabe 2: Dynamische Programmierung

Nachdem nun einige Städte und Dörfer Schulen erhalten haben, will die schleswig-holsteiner Kultusministerin diese besuchen. Dabei möchte Sie jedoch möglichst wenig Zeit brauchen und damit auch möglichst wenig Kilometer zurücklegen. Gesucht ist also ein Weg, der sie alle Schulen besuchen lässt und zudem minimal ist.

- Die einzelnen Schulen sind mit 1..n gekennzeichnet.
 - $D=(d_{ij})$ sei eine Matrix, die die Entfernungen der Schulen zueinander beinhaltet.
 - Der gesuchte Weg startet und endet in Schule 1.
- a) Welche Laufzeit hätte ein Brute-Force Algorithmus, bei dem alle möglichen Wege durchprobiert werden würden? Begründen Sie die Laufzeit. (2 Punkte)
- b) Geben Sie eine Lösung mittels dynamischer Programmierung für dieses Problem an:
- (i) Was ist das geeignete Teilproblem, auf dass sich die Suche nach dem optimalen Weg zurückführen lässt? (2 Punkte)
 - (ii) Geben Sie den gesamten Algorithmus zur Lösung des Problems in Pseudo-Code an. (6 Punkte)
 - (iii) Was ist die worst-case Laufzeit Ihres Algorithmus? (1 Punkt)
 - (iv) Wo liegt der Unterschied ihres Algorithmus zum Greedy Algorithmus aus der Vorlesung? (1 Punkt)

Abgabetermin: Donnerstag den 23. April, bis 11 Uhr Abgabekasten IFIS