

Berechnung kürzester Wege

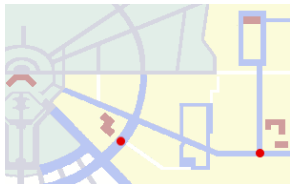
7. Algorithmus der Woche
Informatikjahr 2006

Prof. Dr. Peter Sanders
Dipl.-Inform. Johannes Singler

18. April 2006

Gegeben

- ein Stadtplan
 - ein Startpunkt
 - ein Zielpunkt

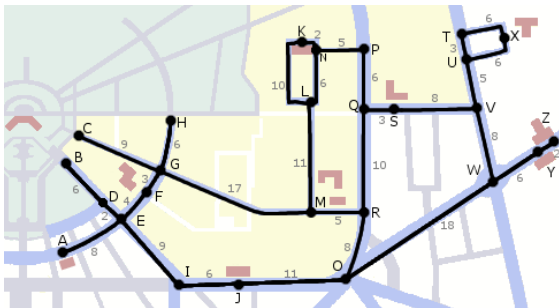


Frage

Wie finde ich den kürzesten Weg zwischen Start- und Zielpunkt?

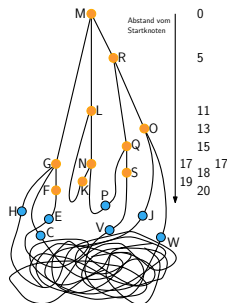
Trickreiche Lösung: Vorbereitung

- Lege den Stadtplan auf den Tisch
- Lege Bindfäden entlang der Straßen
- Verknote die Fäden an Kreuzungen und Abzweigungen
 - zusätzliche Knoten an möglichen Start- und Zielpunkten sowie Abzweigungen
- Markiere die Knoten eindeutig



Trickreiche Lösung: Lösungsfindung

- Fasse das Gespinst am Startpunkt (2)
- Ziehe diesen Knoten langsam nach oben
 - Knoten für Knoten löst sich von der Tischplatte
 - Stoppe, wenn Zielknoten in der Luft
- Mache Zielpunkt ausfindig (5)
- Gehe straff gespannte Fäden zurück
 - führt eindeutig zum Startpunkt
 - Folge von Knoten ist kürzester Weg
- Messe die Entfernung mit Maßband an straffem Weg



- Warum ist der gefundene Weg der kürzeste?
 - Start und Ziel hätten sich sonst nie so weit voneinander entfernt
- Nachteile dieser Lösung
 - aufwändig
 - unpräzise
 - automatisierbar
- ⇒ Der Computer soll die Aufgabe übernehmen!

- „Algorithmus von Dijkstra“ (1959)
- Berechnet kürzeste Wege vom Startknoten zu allen anderen (erreichbaren) Knoten

Eingabe

- virtuelles Netz mit
 - Knoten
 - Fäden mit Längenangaben

Ausgabe

- Länge des kürzesten Wegs vom Startknoten aus zu jedem Knoten
- Teilmenge Fäden zum Rückverfolgen der kürzesten Wege

Alle Knoten liegen, alle $d[]$ sind unendlich, nur $d[\text{Startknoten}] = 0$

while *es gibt liegende Knoten* **do**

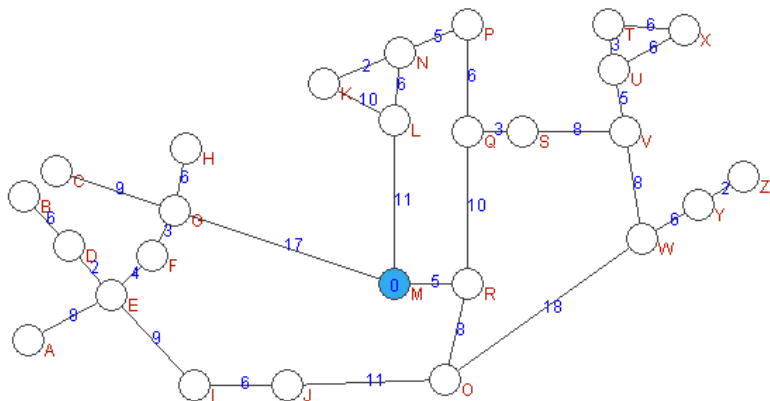
$v :=$ der liegende Knoten mit kleinstem $d[v]$

Mache v hängend

forall *Fäden von v zu einem Nachbar u der Länge ℓ* **do**

if *$d[v] + \ell < d[u]$* **then** *$d[u] := d[v] + \ell$*

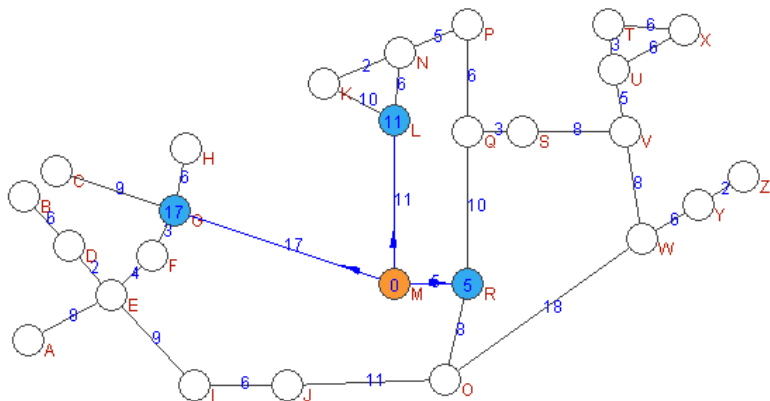
// kürzeren Weg hin zu u gefunden, führt über v



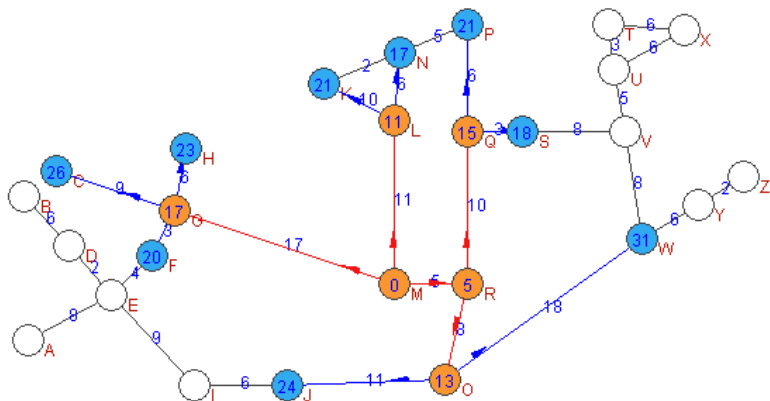
- Nullter Schritt des Algorithmus
 - alle Knoten liegen, Startknoten M blau

Implementierung: Beispiellauf

sk



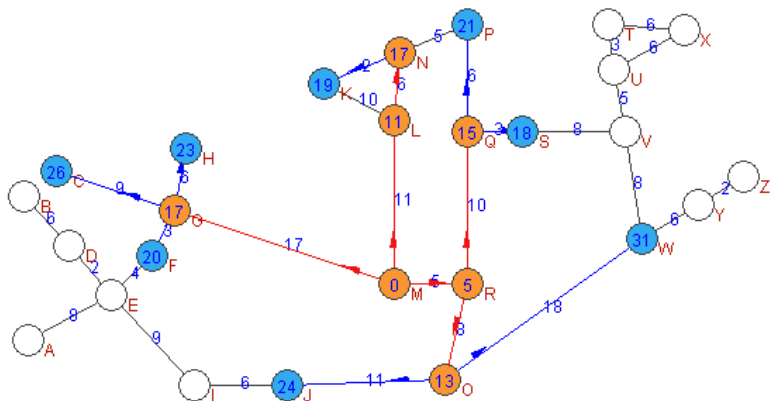
- Erster Schritt des Algorithmus
 - nur M angehoben (orange), nächste Kandidaten blau



- nach 6 Schritten des Algorithmus
 - Der kürzeste Weg zu K geht *noch* von L direkt nach K.

Implementierung: Beispiellauf

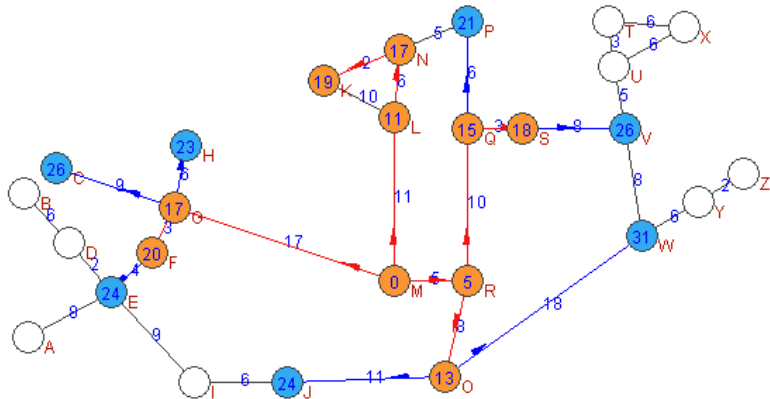
8
k



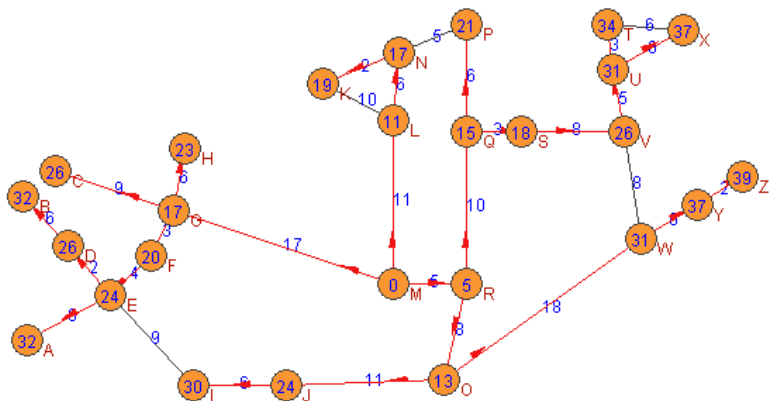
- nach 7 Schritten des Algorithmus
 - Der kürzeste Weg zu K geht jetzt über N.

Implementierung: Beispiellauf

8k



- nach 10 Schritten des Algorithmus
 - Knoten F ist erreicht.



- kompletter Durchlauf des Algorithmus
 - Alle Knoten sind besucht, endgültige Distanzen stehen fest.

- Problem: Finde kürzesten Weg von Startpunkt zu Zielpunkt in einem gegebenen Straßennetz.
- Lösung: Der Algorithmus von Dijkstra berechnet die kürzesten Wege von einem Startpunkt zu allen erreichbaren Zielpunkten.
 - arbeitet analog zur trickreichen Lösung mittels Bindfäden

- weitere Informationen unter

<http://www-il.informatik.rwth-aachen.de/~algorithmus/>

<http://algo2.iti.uni-karlsruhe.de/adw.php>