Vorlesung Theoretische Informatik

Prof. Dr. Martin Schmollinger

Fakultät für Informatik
Masterstudiengang Wirtschaftsinformatik
überarbeitet von F. Laux (Stand: 19.06.2010)

Sommersemester 2010

Hochschule Reutlingen
Reutlingen University

Foliensatz - Inhaltsverzeichnis

1. Algorithmen auf gewichteten Graphen
2. Kürzeste Wege
    - Dijkstra Algorithmus

Algorithmen auf gewichteten Graphen

Erweiterung des einfachen Graphenmodells

- Zuordnung eines Kantengewichts je Kante
- Anwendungen für gerichtete und ungerichtete Graphen

Beispiele

- Flugverbindungen mit Meilenangaben (ungerichtet)
- Straßennetze mit Kilometerangaben (gerichtet)
- Internet mit Übertragungsgeschwindigkeit (gewichtet)

Kürzeste Wege

Gewichtsfunktion

Ein gewichteter Graph wird als G=(V,E,g) mit einer Gewichtsfunktion $g : E \to \mathbb{N}$ notiert, wenn wir als Kantengewichte natürliche Zahlen zulassen wollen.

Gewicht bzw. Länge

Das Gewicht w oder auch die Länge eines Pfades P wird durch Aufsummieren der Kantengewichte bestimmt.

$$w(P) = \sum_{i=1}^{n-1} g((v_i, v_{i+1}))$$

Kürzeste Wege

Distanz

Die Distanz zweier Punkte d(u,v) ist das Gewicht des kürzesten Pfades von u nach v.

Bemerkungen

- Es kann mehrere Pfade zwischen zwei Knoten mit gleicher Länge geben.
- Kürzeste Wege müssen nicht existieren!

Dijkstra Algorithmus

Zum Algorithmus

- Sehr bekannter Graphenalgorithmus, der von Dijkstra 1959 veröffentlicht wurde
- Weiterentwicklung der Breitensuche für gewichtete Kanten
- Funktioniert nur für nicht-negative Kanten

Dijkstra Algorithmus

Pseudo-Code

```
algorithm Dijkstra (G,s)
  Eingabe Graph G (V,E,g) mit Startknoten s
  for each Knoten u aus V - s do
     D[u]:=∞;
  od;
  D[s]:=0; PriorityQueue Q:=V;
  while not Q.isEmpty() do
     u:=Q.extractMinimal(); // Q := Q \ u
     for each v aus ZielknotenAusgehenderKanten(u) ⋂ Q do
       if D[u]+g((u,v))<D[v] then
         adjustiere Q an neuen Wert D[v];
       fi
     od;
  od;
```

Überlegungen zur Korrektheit

Vorüberlegungen

- Es liegen nur nicht-negative Kantengewichte vor.
- Betrachten wir einen Iterationschritt und nehmen an, dass wir in den bisherigen Iterationen jeweils einen Knoten mit korrektem Distanzwert hinzugenommen haben.
- Im Iterationsschritt wird die „billigste“ Verbindung zu einem noch nicht bearbeiteten Knoten hinzugenommen.

Folgerungen

- Der neue Distanzwert wird durch den „billigsten“ aus dem bisher bearbeiteten Teilgraphen um genau eine Kante hinausgehenden Pfad bestimmt.
- Jeder längere Pfade zum gleichen Ziel ist teuerer, da Kosten nicht sinken.
- Bei mehreren Pfaden zu einem noch nicht bearbeiteten Knoten garantiert die Dreiecksungleichung die kürzeste Distanz.

Laufzeit des Dijkstra-Algorithmus

Die Laufzeit ist abhängig von der Realisierung der Prioritätswarteschlange!

Allgemeine Formel

$O(nT_{extractMin} + mT_{decreaseKey})$

Annahme: Alle Operationen in O(1)

$O(n + m)$

Übung macht den Meister!

Lineare Liste realisiert PQ

Wie ist die Laufzeit bei Verwendung einer linearen Liste?

Alternativen?

Überlegen Sie sich geeignetere Realisierungen der Priority Queue!