Vortrag gehalten am 11. Mai 2004
Zeitansatz 30min
Otto-von-Guericke-Universität Magdeburg

Algorithmen auf Graphen kürzeste Wege

Ein Vortrag von Markus Durzinsky

Student der Otto-von-Guericke-Universität Magdeburg

www-e.uni-magdeburg.de/durzinsk
Mardur@gmx.net

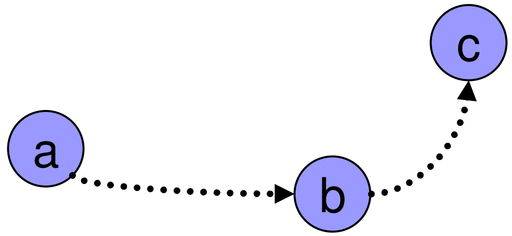
Gliederung

1. Problem der kürzesten Wege
2. Dijkstra-Algorithmus
3. Floyd-Warshall-Algorithmus
4. Zusammenfassung
5. Literatur

Problem der kürzesten Wege

- gesucht ist ein Weg
  - zusammenhängende Folge von Kanten
  - ohne Wiederholungen
- mit minimalem Gewicht
- Problem ist NP-schwer
  - längste Wege mit negativem Kantengewicht
  - ↳ Lösung des Hamilton-Kreis-Problems

Allgemeiner Lösungsansatz



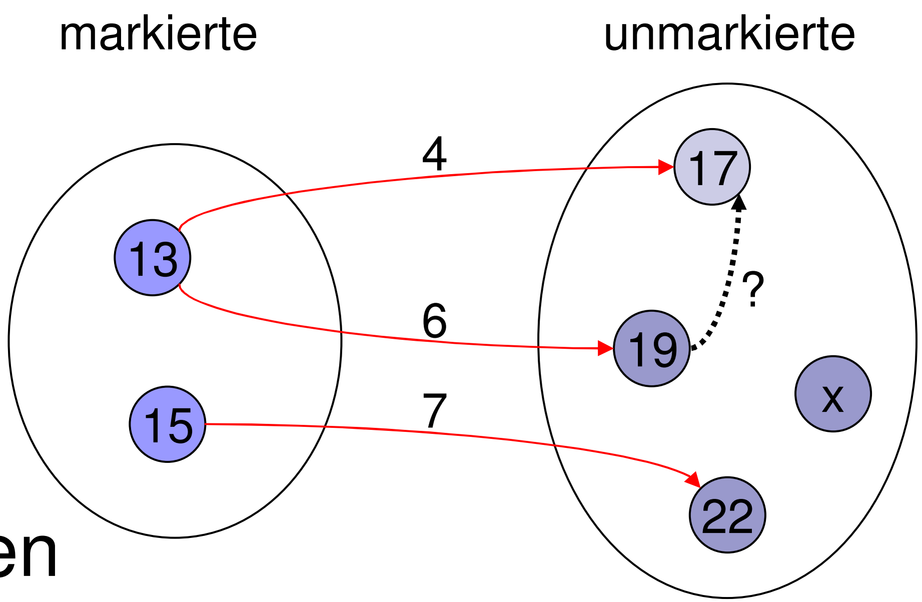
- kürzester Weg von a nach c besteht aus
  - kürzestem Weg von a nach b
  - kürzestem Weg von b nach c
- Aufbau eines Algorithmus

```
wähle Knoten a,b,c
wenn Weg(a,c) länger als Weg(a,b) + Weg(b,c) dann
  setze Weg(a,c) auf Weg(a,b) + Weg(b,c)
```

- Unterschied nur in der Wahl der Knoten

Dijkstra-Algorithmus

- zu markierten Knoten ist ein kürzester Weg bekannt
- Wann kann sich der bisherige Abstand unmarkierter Knoten verkürzen?



↳ markiere unmarkierten Knoten mit geringstem Abstand

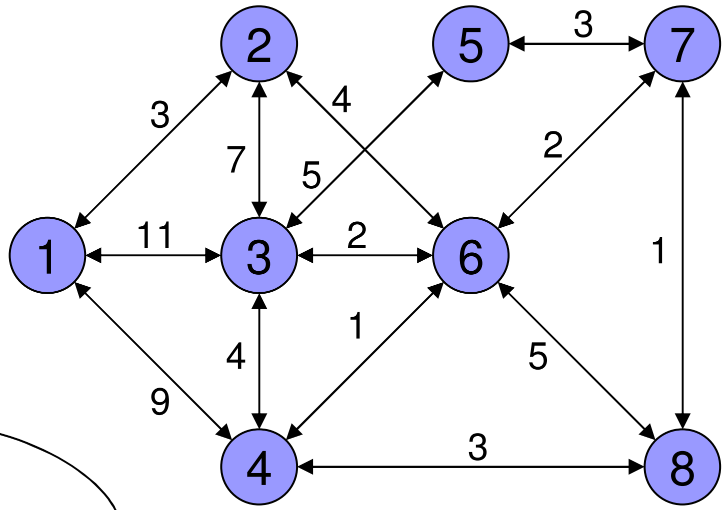
Notiz: Problem bei negativen Kanten

Dijkstra-Algorithmus

```
Input: G =(V,E), c(e) >= 0, s∈V
Output: Aboreszenzbaum (pre,dist)

pre[v]  := s       ∀ v∈V
dist[v] := infty   ∀ v∈V
dist[s] := 0

while V≠∅ do
  wähle v∈V mit c(v) minimal
  if dist[v] = infty then
    return
  V := V \ {v}
  forall (v,u)∈E do
    if dist[u] > dist[v] + c(v,u) then
      dist[u] := dist[v] + c(v,u)
      pre[u]  := v
```

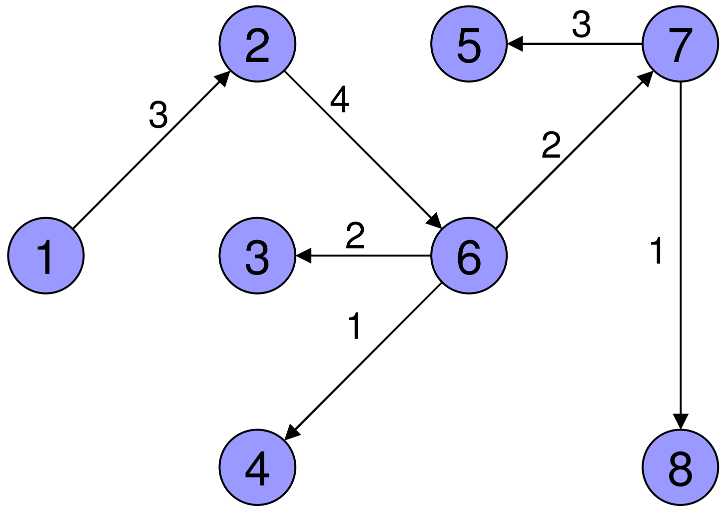


zeitkritisch !!

Notiz: Laufzeitvergleich Lineare Suche vs. Heap

Lösung für Startknoten 1

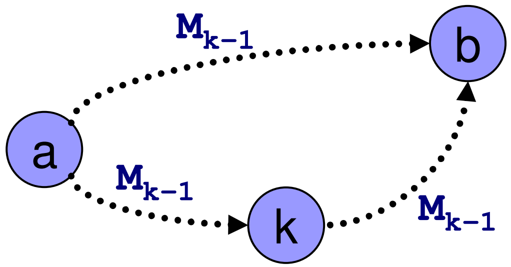
- alle kürzesten Wege von einem Startknoten
- Kürzester-Wege-Baum
  auch Aboreszenz



v	1	2	3	4	5	6	7	8
pre[v]	1	1	6	6	7	2	6	7
dist[v]	0	3	9	8	12	7	9	10

Eine andere Variante

- Beschränkung der inneren Knoten eines Weges auf eine kleinere Menge
- Iteration über die Größe der Menge



$V = \{1, 2, 3, 4, ..., n\}$

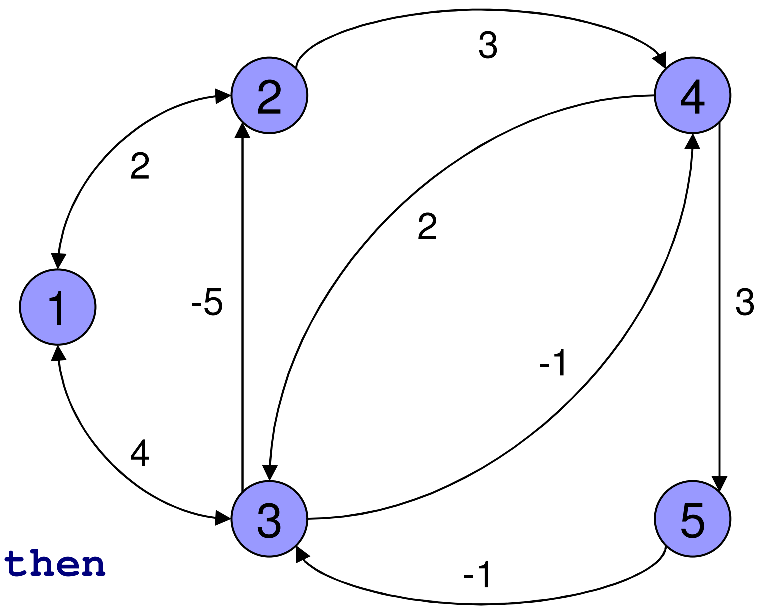
$M_1 = \{1\}$ Wege nur über Knoten 1
$M_2 = \{1, 2\}$ Wege aus M_1 und über Knoten 2
...
$M_k = \{1, ..., k\}$ Wege aus M_{k-1} und über Knoten k

Floyd-Warshall-Algorithmus

```
Input:  c(u,v) ∀ u,v ∈ V, V={1,2,..,n}
        keine negativen Zyklen
Output: d(u,v) – Länge des Weges
         e(u,v) – höchster innerer
                  Knoten

e(u,v) := 0         ∀ u,v ∈ V
d(u,v) := c(u,v)    ∀ u,v ∈ V
d(u,u) := infty     ∀ v ∈ V

for k := 1 to n do
  for u := 1 to n do
    for v := 1 to n do
      if d(u,v) > d(u,k) + d(k,v) then
        d(u,v) := d(u,k) + d(k,v)
        e(u,v) := k
```



Notiz: Probleme bei negativen Zyklen

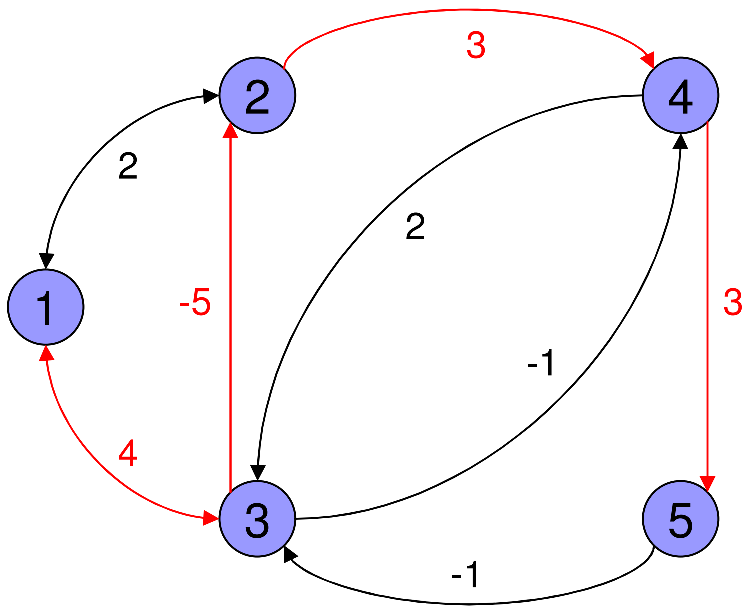
Rekonstruktion eines Weges

- Suche Weg von Knoten 1 nach Knoten 5

```
d =  1  -1   4   2   5
     2   0   5   3   6
    -3  -5   0  -2   1
    -1  -3   2   0   3
    -4  -6  -1  -3   0

e =  3   3   0   3   4
     0   4   4   0   4
     2   0   4   2   4
     3   3   0   3   0
     3   3   0   3   4

          1 - 5
        1 - 4 - 5
      1 - 3 - 4 - 5
    1 - 3 - 2 - 4 - 5
```



Zusammenfassung

- kürzeste Wege mit beliebigen Kantengewichten sind schwer zu berechnen
- effektive Algorithmen nur für spezielle Graphen
- **Dijkstra**, falls keine negativen Kanten
  - gieriges Ausbreiten einer Suchfront
- **Floyd-Warshall**, falls keine negativen Kreise
  - schrittweise Erweiterung der Zwischenknoten

Literatur / Links

- Peter Gritzmann, Rene Brandenburg: *Das Geheimnis des kürzesten Weges*
- Kyle Loudon*: Algorithmen mit C*
- Internet
  - Vorlesung Computerorientierte Mathematik
    `http://www.math.uni-magdeburg.de/~mkoeppe/lehre/coma-2003`