Der Algorithmus von Dijkstra

(Berechnung kürzester Wege in bewerteten Graphen)

GIS Praxis II, Jan Hinzmann, Matr.-Nr.: 2068095

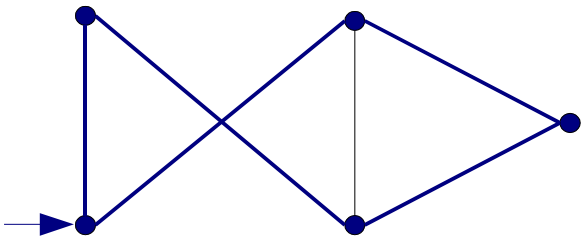
Inhaltsverzeichnis

Edsger Wybe Dijkstra

- Holländer (1930 Rotterdam– 2002)
- erster Programmierer der Niederlande
- 1972 Turing Preis
- *"In der Informatik geht es genau so wenig um Computer wie in der Astronomie um Teleskope."*
- Dijkstra-Algorithmus
  - findet kürzeste Wege in *Graphen*

Graphen

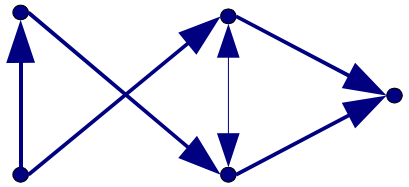
- Ein Graph G(N,E) hat
  - N *(Node)*: Menge der *Knoten*
  - E *(Edge)*: Menge der *Kanten*, die Knoten verbinden
  - Knoten $N := \{n_1, n_2, ..., n_n\}$
  - Kanten $E := \{e_1(n_i, n_j), ..., e_2(n_k, n_l)\}$



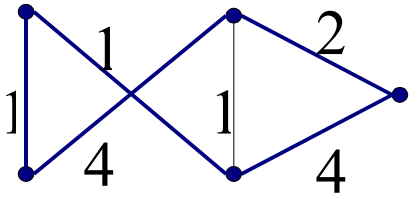
- es gibt *gerichtete*, *gewichtete*, *benannte*, ... Graphen

Graphen (2)

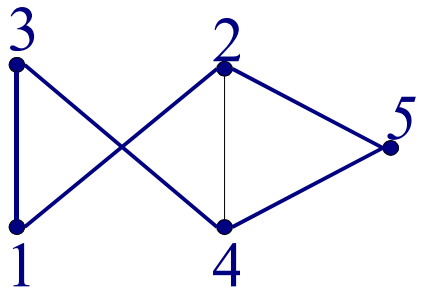
- Ein *gerichteter* Graph
  - gerichtete Kanten



- Ein *gewichteter* Graph
  - gewichtete Kanten

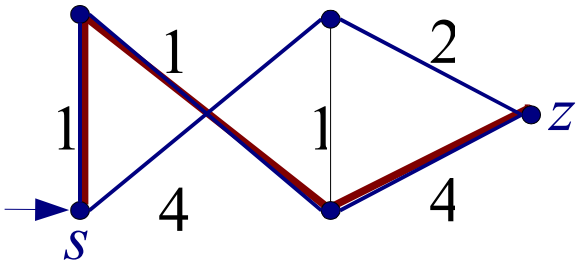


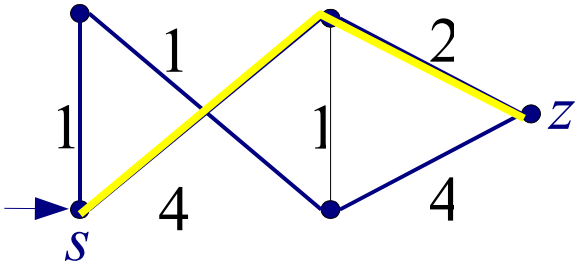
- Ein *benannter* Graph
  - benannte Knoten

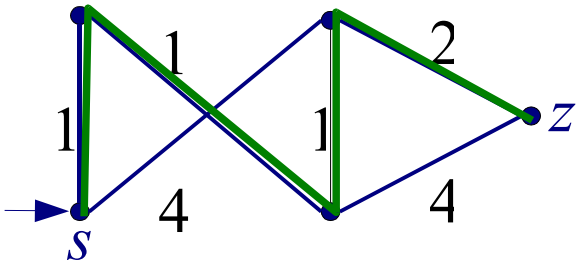


- *kurze Wege* in Graphen?

Kürzester Weg?







- Oft gibt es mehrere Wege:
  - Was ist der kürzeste Weg vom Startknoten s zum Ziel z ?
  - **Weg 1** kostet 9
  - **Weg 2** kostet 6
  - **Weg 3** aber nur 5 !

- Die Lösung berechnet der Algorithmus von Dijkstra ...

Algorithmus


```
Knotenmenge s,k; //s = Startknoten
Knotenmenge opti = {s};
Knotenmenge rest = k\{s};

for ( k aus rest ) do
      D[k] = d[s,k] (es gibt Weg);
           = ∞ sonst;
done;

while ( rest nicht leer ) do;
      wähle k aus rest mit min(D[k]);
      opti += {k};
      rest = rest -{k};
      for ( alle Knoten n von k ) do
            D[n] = min(D[n], D[k] + d[k,n]);
      done;
done;
```

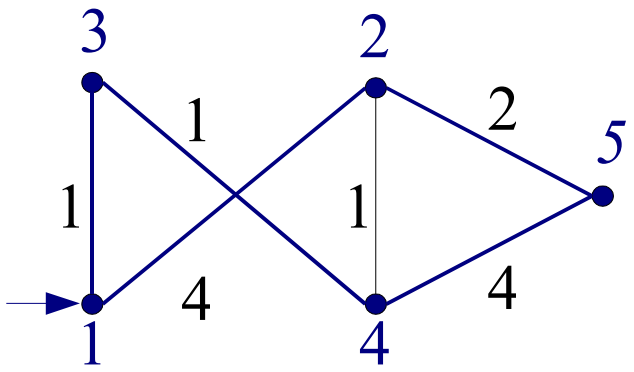
Initialisierung

Iteration

In Worten:

1. Der Startknoten kommt in *opti*
2. Alle anderen Knoten in *rest*
3. berechne alle Distanzen für die Knoten in *rest*
4. verschiebe den Knoten mit der kleinsten Distanz von *rest* nach *opti*
5. berechne die Distanzen für die Knoten in *rest* neu
6. wiederhole Schritt 4 und 5 solange, bis *rest* leer ist

Beispiel



Opt	D[2]	D[3]	D[4]	D[5]	Rest
{1}	4	**1**	∞	∞	{2,3,4,5}
{1,3}	4	-	**2**	∞	{2,4,5}
{1,3,4}	**3**	-	-	6	{2,5}
{1,3,4,2}	-	-	-	**5**	{5}
{1,3,4,2,5}	-	-	-	-	Ø

- Der Dijkstra-Algorithmus findet die kürzesten Distanzen zu allen anderen Knoten, wenn keine negativen Distanzen (Betrag) zugelassen sind
- Für die Navigation kann er abgebrochen werden, sobald der Zielknoten zu Opt hinzugefügt worden ist

Applet

- Dijkstra Applet
- http://carbon.cudenver.edu/~hgreenbe/sessions/dijkstra/DijkstraApplet.html

Diskussion

- Edsger Wybe Dijkstra
- Graphen (gerichtet, gewichtet)
- Kürzester Weg?
- Algorithmus
- Beispiel
- Applet

– Danke –