



Einführung in minimale Spann­bäume und deren Berechnung

Johannes Diemke

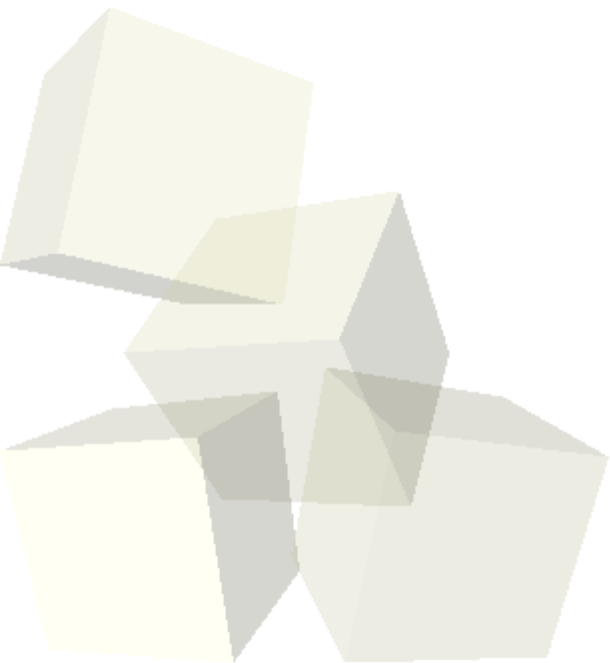
Nils Sommer

Universität Oldenburg, FK II – Department für Informatik
Vortrag im Rahmen des Proseminars 2006



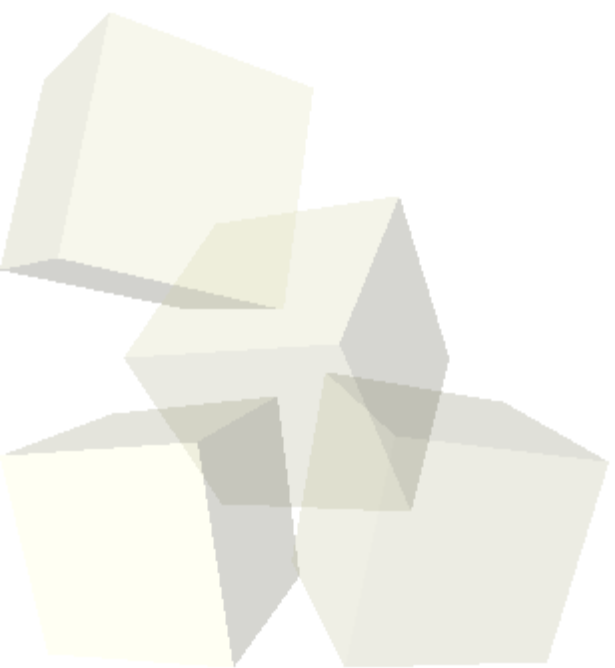
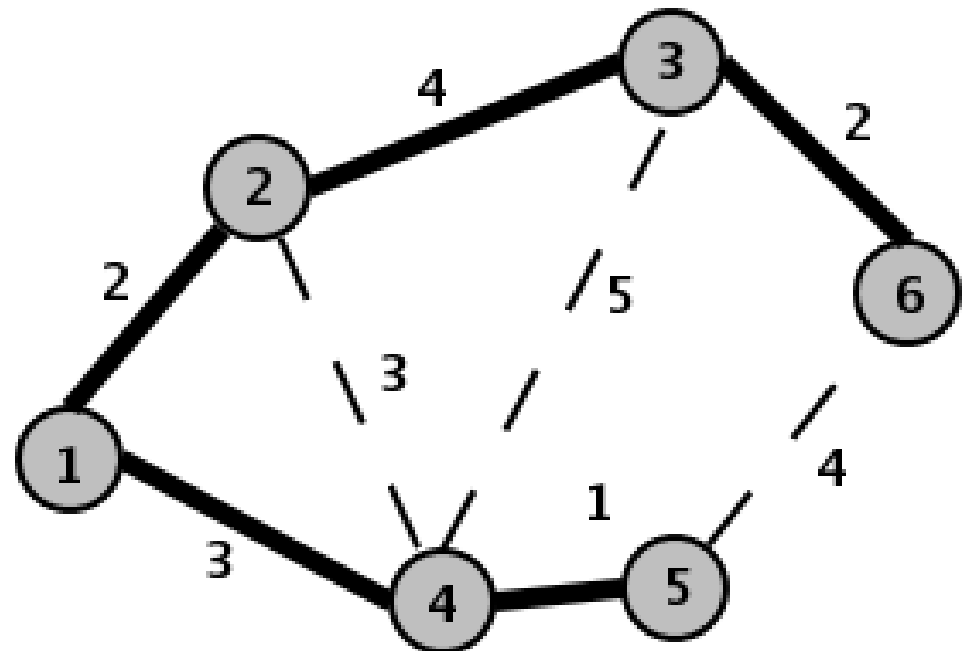


- Spannbäume
- Greedy Algorithmen
- Algorithmus von Kruskal
- Prims Algorithmus
- Praktische und theoretische Bedeutung
- Resümee





- Spannbaum $T=(V,E')$
- Teilgraph eines ungerichteten zusammenhängenden Graphen $G=(V,E)$
- Mit $E' \subset E$
- T ist ein Baum (keine Zyklen)





- Kantengewichtete Graphen $G=(V,E,w)$
- Zuordnung einer reellen Zahl zu jeder Kante
- Gewichtungsfunktion $w:E \Rightarrow \mathbb{R}$
- Gewicht des Graphen

$$G = \sum_{e \in E} w(e)$$

- Ein Spannbaum ist minimal gdw. kein Spannbaum mit geringerem Gewicht existiert





Algorithmus von Kruskal

- Konstruiert einen minimalen Spannbaum $T=(V,E')$ aus einem zusammenhängenden, ungerichteten, gewichteten Graphen $G=(V,E)$
- Funktionsweise:
 - ♦ T enthält genau die Knoten aus G
 - ♦ Setze $E'=\{\}$
 - ♦ Füge die Kanten sukzessiv in aufsteigender Reihenfolge ihres Gewichtes in Abhängigkeit einer Auswahlbedingung zu E' hinzu





Algorithmus von Kruskal

■ Auswahlbedingung:

- ♦ Führt das Einfügen einer Kante in T zu Zyklen, so verwerfe die Kante. Ansonsten füge sie ein.

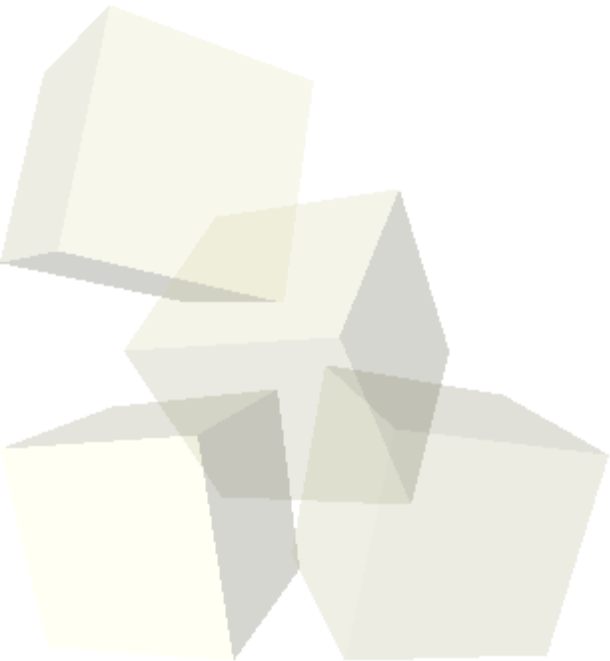
■ Pseudocode:

```
while(!isConnectedGraph(T)) {  
  
    edge = getEdgeWithMinimumCost(E);  
  
    deleteEdgeFromGraph(E, edge);  
  
    if(!addingEdgeToGraphCreatesCycle(T, edge)) {  
        addEdgeToGraph(T, edge)  
    }  
}
```



Algorithmus von Kruskal

- Konkrete Implementierungen nutzen:
 - ◆ Priority-Queue
 - ◆ Merge-Find-Struktur
- Hauptaufwand durch Verwaltung der Priority-Queue
- Insgesamt ergibt sich ein Gesamtaufwand von $O(e \log e)$





■ Schritte:

- ♦ Wähle beliebigen Knoten
- ♦ Füge Kante hinzu, die einen neuen Knoten aus dem Ausgangsgraphen mit dem neuen Graphen verbindet.
- ♦ Kante zu diesem Knoten muss minimale Gewichtungen haben
- ♦ Fertig wenn Kantenmenge von beiden Graphen gleich



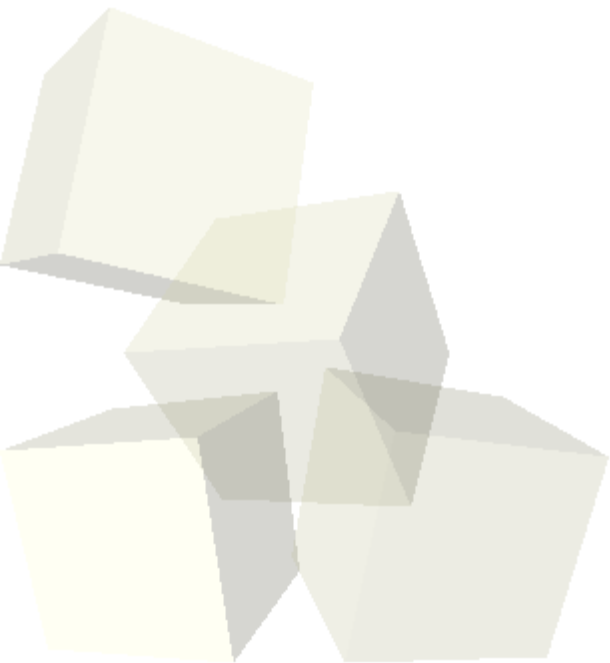


■ Pseudocode:

```
public static Graph primMinSpanTree(Graph
input) {
    Graph T = new Graph();
    T.addNode(input.getAnyNode());
    while(!T.nodesEqual(input)) {
        Edge edges[] =
            getEdgesWhichAddNewNode(input, T);
        edges.sortByWeight();
        T.addEdgeWithNode(edges[0]);
    }
    return T;
}
```



- Noch stark optimierbar
 - ◆ Datenstrukturen
 - ◆ Sortieralgorithmus
- In diesem Fall $O(n^2)$
- Immer optimal



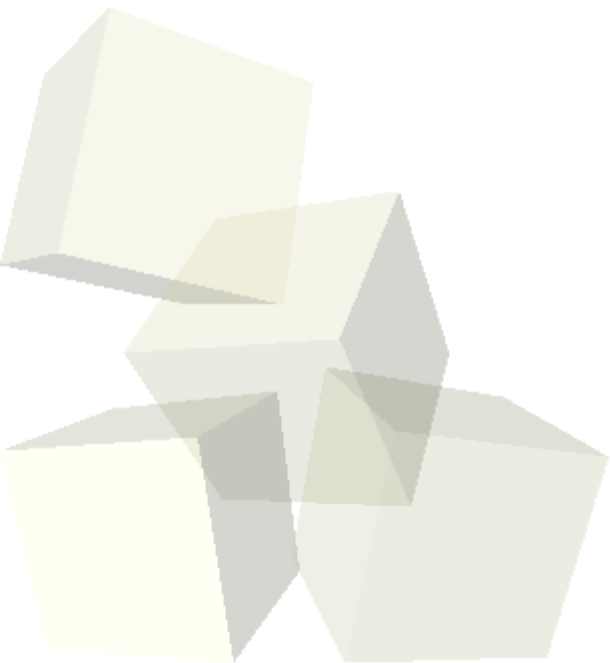
Praktische und Theoretische Bedeutung

■ Theorie

- ◆ Lehre
- ◆ Basis für komplexere Probleme

■ Praxis:

- ◆ Netzwerke: Computer, Elektrische Netze
- ◆ Routing





- Weitgehend erforschtes Gebiet
- Forschung schon in den 50er Jahren (Bell)
- In der Praxis oft eingesetzt

