

Graphs: Minimum Spanning Trees (Chapter 9)

CSE 373
Data Structures and Algorithms

11/29/2010 1

Today's Outline

- Announcements
 - Homework #6/7 – Partner Selection due Wed (11:45pm).
- Graphs
 - Shortest Paths Algorithms
 - Minimum Spanning Tree

11/29/2010 2

Minimum Spanning Trees

Given an undirected graph $G=(V,E)$, find a graph $G'=(V, E')$ such that:

- E' is a subset of E
- $|E'| = |V| - 1$
- G' is connected
- $\sum_{(u,v) \in E'} c_{uv}$ is minimal

G' is a **minimum spanning tree.**

Applications: wiring a house, power grids, Internet connections

11/29/2010 3

Student Activity

Find the MST

11/29/2010 4

Two Different Approaches

Prim's Algorithm
Almost identical to Dijkstra's

Kruskal's Algorithm
Completely different!

11/29/2010 5

Prim's algorithm

Idea: Grow a tree by adding an edge from the "known" vertices to the "unknown" vertices. Pick the edge with the smallest weight.

11/29/2010 6

Prim's Algorithm for MST

A node-based greedy algorithm
Builds MST by greedily adding nodes

- Select a node to be the "root"
 - mark it as known
 - Update cost of all its neighbors
- While there are unknown nodes left in the graph
 - Select an unknown node b with the smallest cost from some known node a
 - Mark b as known
 - Add (a, b) to MST
 - Update cost of all nodes adjacent to b

11/29/2010 7

Student Activity Start with V_1

Find MST using Prim's

V	Kwn	Distance	path
v1			
v2			
v3			
v4			
v5			
v6			
v7			

Order Declared Known:
 V_1

11/29/2010 8

Prim's Algorithm Analysis

Running time:
Same as Dijkstra's: $O(|E| \log |V|)$

Correctness:
Proof is similar to Dijkstra's

11/29/2010 9

Kruskal's MST Algorithm

Idea: Grow a forest out of edges that do not create a cycle. Pick an edge with the smallest weight.

$G=(V,E)$

11/29/2010 10

Kruskal's Algorithm for MST

An edge-based greedy algorithm
Builds MST by greedily adding edges

- Initialize with
 - empty MST
 - all vertices marked unconnected
 - all edges unmarked
- While there are still unmarked edges
 - Pick the lowest cost edge (u, v) and mark it
 - If u and v are not already connected, add (u, v) to the MST and mark u and v as connected to each other

Doesn't it sound familiar?

11/29/2010 11

Kruskal code

```

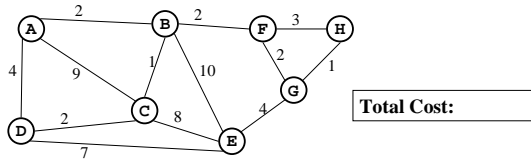
void Graph::kruskal(){
  int edgesAccepted = 0;
  DisjSet s(NUM_VERTICES);

  while (edgesAccepted < NUM_VERTICES - 1){
    e = smallest weight edge not deleted yet;
    // edge e = (u, v)
    uset = s.find(u);
    vset = s.find(v);
    if (uset != vset){
      edgesAccepted++;
      s.unionSets(uset, vset);
    }
  }
}
  
```

|E| heap ops
2|E| finds
|V| unions

11/29/2010 12

Find MST using Kruskal's



- Now find the MST using Prim's method.
- Under what conditions will these methods give the same result?