CSE 326: Data Structures

Spanning Trees

James Fogarty

Spring 2009
• Recall that trees are not always rooted
Spanning Tree in a Graph

Vertex = router
Edge = link between routers

Spanning tree
- Connects all the vertices
- No cycles
• **G = (V,E)**
  - V is a set of vertices (or nodes)
  - E is a set of unordered pairs of vertices

V = \{1,2,3,4,5,6,7\}

E = \{\{1,2\},\{1,6\},\{1,5\},\{2,7\},\{2,3\},\{3,4\},\{4,7\},\{4,5\},\{5,6\}\}
Spanning Tree Problem

• Input: An undirected graph $G = (V,E)$ such that $G$ is connected.

• Output: $T$ contained in $E$ such that
  – $(V,T)$ is a connected graph
  – $(V,T)$ is acyclic
How Could We Find a Spanning Tree?

V = \{1, 2, 3, 4, 5, 6, 7\}

E = \{\{1, 2\}, \{1, 6\}, \{1, 5\}, \{2, 7\}, \{2, 3\}, \{3, 4\}, \{4, 7\}, \{4, 5\}, \{5, 6\}\}
Spanning Tree Algorithm

ST(i: vertex)
  mark i;
  for each j adjacent to i do
    if j is unmarked then
      Add {i,j} to T;
      ST(j);
  end{ST}

Main
T := empty set;
ST(1);
end{Main}
Example of Depth First Search

ST(1)
Example of Depth First Search

{1, 2}
Example of Depth First Search

ST(1)
ST(2)
ST(7)

{1,2} {2,7}
Example of Depth First Search

ST(1)
ST(2)
ST(7)
ST(5)

\{1,2\} \{2,7\} \{7,5\}
Example of Depth First Search

\{1,2\} \{2,7\} \{7,5\} \{5,4\}

ST(1)
ST(2)
ST(7)
ST(5)
ST(4)
Example of Depth First Search

ST(1)
ST(2)
ST(7)
ST(5)
ST(4)
ST(3)

{1,2} {2,7} {7,5} {5,4} {4,3}
Example of Depth First Search

\{1,2\} \{2,7\} \{7,5\} \{5,4\} \{4,3\}

ST(1)
ST(2)
ST(7)
ST(5)
ST(4)
ST(3)
Example of Depth First Search

\{1,2\} \{2,7\} \{7,5\} \{5,4\} \{4,3\}

ST(1)  
ST(2)  
ST(7)  
ST(5)  
ST(4)  

Graph showing nodes and edges.
Example of Depth First Search

\{{1,2} \{2,7} \{7,5} \{5,4} \{4,3}\}
Example of Depth First Search

ST(1)
ST(2)
ST(7)
ST(5)

{1,2} {2,7} {7,5} {5,4} {4,3}
Example of Depth First Search

\{1,2\} \{2,7\} \{7,5\} \{5,4\} \{4,3\}

ST(1)
ST(2)
ST(7)
ST(5)
Example of Depth First Search

{1,2} {2,7} {7,5} {5,4} {4,3} {5,6}

ST(1)  ST(2)  ST(7)  ST(5)  ST(6)
Example of Depth First Search

{1,2} {2,7} {7,5} {5,4} {4,3} {5,6}
Example of Depth First Search

\{1,2\} \{2,7\} \{7,5\} \{5,4\} \{4,3\} \{5,6\}
Example of Depth First Search

{1,2} {2,7} {7,5} {5,4} {4,3} {5,6}

ST(1)
ST(2)
ST(7)
Example of Depth First Search

ST(1)  ST(2)

{1,2} {2,7} {7,5} {5,4} {4,3} {5,6}
Example of Depth First Search

\{1,2\} \{2,7\} \{7,5\} \{5,4\} \{4,3\} \{5,6\}
Final Spanning Tree

{1,2} {2,7} {7,5} {5,4} {4,3} {5,6}
Best Spanning Tree

Finding a reliable routing subnetwork:
• Each edge has the probability that it won’t fail
• Find the spanning tree that is least likely to fail
Example of a Spanning Tree

Probability of success = 0.85 \times 0.95 \times 0.89 \times 0.95 \times 1.0 \times 0.84
= 0.5735
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.
Reducing Best to Minimum

Let $P(e)$ be the probability that an edge doesn’t fail. Define:

$$C(e) = -\log_{10}(P(e))$$

Minimizing $\sum_{e \in T} C(e)$

is equivalent to maximizing $\prod_{e \in T} P(e)$

because $\prod_{e \in T} P(e) = \prod_{e \in T} 10^{-C(e)} = 10^{-\sum_{e \in T} C(e)}$
Example of Reduction

Best Spanning Tree Problem

Minimum Spanning Tree Problem
Find the MST
Find the MST
Two Different Approaches

Prim’s Algorithm
Looks familiar!

Kruskals’s Algorithm
Completely different!
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.
Prim’s Algorithm for MST

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

1. Select a node to be the “root”
   • mark it as known
   • Update cost of all its neighbors
2. While there are unknown nodes left in the graph
   a. Select an unknown node $b$ with the smallest cost to reach from some known node $a$
   b. Mark $b$ as known
   c. Add $(a, b)$ to MST
   d. Update cost of all nodes adjacent to $b$
Find MST using Prim’s

<table>
<thead>
<tr>
<th>V</th>
<th>Kwn</th>
<th>Distance</th>
<th>path</th>
</tr>
</thead>
<tbody>
<tr>
<td>v1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>v2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>v3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>v4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>v5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>v6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>v7</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Start with $V_1$

Order Declared Known:
Find MST using Prim’s

Start with $V_1$

Order Declared Known:

$V_1$

<table>
<thead>
<tr>
<th>V</th>
<th>Kwn</th>
<th>Distance</th>
<th>path</th>
</tr>
</thead>
<tbody>
<tr>
<td>v1</td>
<td>Y</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>v2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>v3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>v4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>v5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>v6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>v7</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Find MST using Prim’s

Start with $V_1$

<table>
<thead>
<tr>
<th>V</th>
<th>Kwn</th>
<th>Distance</th>
<th>path</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_1$</td>
<td>Y</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$v_2$</td>
<td></td>
<td>2</td>
<td>$V_1$</td>
</tr>
<tr>
<td>$v_3$</td>
<td></td>
<td>4</td>
<td>$V_1$</td>
</tr>
<tr>
<td>$v_4$</td>
<td></td>
<td>1</td>
<td>$V_1$</td>
</tr>
<tr>
<td>$v_5$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$v_6$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$v_7$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Find MST using Prim's

Start with $V_1$

<table>
<thead>
<tr>
<th>V</th>
<th>Kwn</th>
<th>Distance</th>
<th>path</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_1$</td>
<td>Y</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$v_2$</td>
<td></td>
<td>2</td>
<td>$V_1$</td>
</tr>
<tr>
<td>$v_3$</td>
<td></td>
<td>4</td>
<td>$V_1$</td>
</tr>
<tr>
<td>$v_4$</td>
<td>Y</td>
<td>1</td>
<td>$V_1$</td>
</tr>
<tr>
<td>$v_5$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$v_6$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$v_7$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Order Declared Known: $V_1, V_4$
Find MST using Prim’s

Start with \( V_1 \)

<table>
<thead>
<tr>
<th>V</th>
<th>Kwn</th>
<th>Distance</th>
<th>path</th>
</tr>
</thead>
<tbody>
<tr>
<td>v1</td>
<td>Y</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>v2</td>
<td>2</td>
<td>V1</td>
<td></td>
</tr>
<tr>
<td>v3</td>
<td>2</td>
<td>V4</td>
<td></td>
</tr>
<tr>
<td>v4</td>
<td>Y</td>
<td>1</td>
<td>V1</td>
</tr>
<tr>
<td>v5</td>
<td>7</td>
<td>V4</td>
<td></td>
</tr>
<tr>
<td>v6</td>
<td>8</td>
<td>V4</td>
<td></td>
</tr>
<tr>
<td>v7</td>
<td>4</td>
<td>V4</td>
<td></td>
</tr>
</tbody>
</table>

Order Declared Known: V1, V4
Find MST using Prim’s

Start with $V_1$

<table>
<thead>
<tr>
<th>V</th>
<th>Kwn</th>
<th>Distance</th>
<th>path</th>
</tr>
</thead>
<tbody>
<tr>
<td>v1</td>
<td>Y</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>v2</td>
<td>Y</td>
<td>2</td>
<td>V1</td>
</tr>
<tr>
<td>v3</td>
<td></td>
<td>2</td>
<td>V4</td>
</tr>
<tr>
<td>v4</td>
<td>Y</td>
<td>1</td>
<td>V1</td>
</tr>
<tr>
<td>v5</td>
<td></td>
<td>7</td>
<td>V4</td>
</tr>
<tr>
<td>v6</td>
<td></td>
<td>8</td>
<td>V4</td>
</tr>
<tr>
<td>v7</td>
<td></td>
<td>4</td>
<td>V4</td>
</tr>
</tbody>
</table>

Order Declared Known: V1, V4, V2
Find MST using Prim’s

Start with $V_1$

<table>
<thead>
<tr>
<th>V</th>
<th>Kwn</th>
<th>Distance</th>
<th>path</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_1$</td>
<td>Y</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$v_2$</td>
<td>Y</td>
<td>2</td>
<td>$V_1$</td>
</tr>
<tr>
<td>$v_3$</td>
<td>Y</td>
<td>2</td>
<td>$V_4$</td>
</tr>
<tr>
<td>$v_4$</td>
<td>Y</td>
<td>1</td>
<td>$V_1$</td>
</tr>
<tr>
<td>$v_5$</td>
<td></td>
<td>7</td>
<td>$V_4$</td>
</tr>
<tr>
<td>$v_6$</td>
<td></td>
<td>8</td>
<td>$V_4$</td>
</tr>
<tr>
<td>$v_7$</td>
<td></td>
<td>4</td>
<td>$V_4$</td>
</tr>
</tbody>
</table>

Order Declared Known: $V_1, V_4, V_2, V_3$
Find MST using Prim’s

Start with $V_1$

<table>
<thead>
<tr>
<th>V</th>
<th>Kwn</th>
<th>Distance</th>
<th>path</th>
</tr>
</thead>
<tbody>
<tr>
<td>v1</td>
<td>Y</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>v2</td>
<td>Y</td>
<td>2</td>
<td>V1</td>
</tr>
<tr>
<td>v3</td>
<td>Y</td>
<td>2</td>
<td>V4</td>
</tr>
<tr>
<td>v4</td>
<td>Y</td>
<td>1</td>
<td>V1</td>
</tr>
<tr>
<td>v5</td>
<td></td>
<td>7</td>
<td>V4</td>
</tr>
<tr>
<td>v6</td>
<td></td>
<td>5</td>
<td>V3</td>
</tr>
<tr>
<td>v7</td>
<td></td>
<td>4</td>
<td>V4</td>
</tr>
</tbody>
</table>

Order Declared Known: V1, V4, V2, V3
Find MST using Prim’s

Start with $V_1$

<table>
<thead>
<tr>
<th>V</th>
<th>Kwn</th>
<th>Distance</th>
<th>path</th>
</tr>
</thead>
<tbody>
<tr>
<td>v1</td>
<td>Y</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>v2</td>
<td>Y</td>
<td>2</td>
<td>V1</td>
</tr>
<tr>
<td>v3</td>
<td>Y</td>
<td>2</td>
<td>V4</td>
</tr>
<tr>
<td>v4</td>
<td>Y</td>
<td>1</td>
<td>V1</td>
</tr>
<tr>
<td>v5</td>
<td></td>
<td>7</td>
<td>V4</td>
</tr>
<tr>
<td>v6</td>
<td></td>
<td>5</td>
<td>V3</td>
</tr>
<tr>
<td>v7</td>
<td>Y</td>
<td>4</td>
<td>V4</td>
</tr>
</tbody>
</table>

Order Declared Known:
V1, V4, V2, V3, V7
Find MST using Prim’s

Start with $V_1$

<table>
<thead>
<tr>
<th>V</th>
<th>Kwn</th>
<th>Distance</th>
<th>path</th>
</tr>
</thead>
<tbody>
<tr>
<td>v1</td>
<td>Y</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>v2</td>
<td>Y</td>
<td>2</td>
<td>V1</td>
</tr>
<tr>
<td>v3</td>
<td>Y</td>
<td>2</td>
<td>V4</td>
</tr>
<tr>
<td>v4</td>
<td>Y</td>
<td>1</td>
<td>V1</td>
</tr>
<tr>
<td>v5</td>
<td></td>
<td>6</td>
<td>V7</td>
</tr>
<tr>
<td>v6</td>
<td></td>
<td>1</td>
<td>V7</td>
</tr>
<tr>
<td>v7</td>
<td>Y</td>
<td>4</td>
<td>V4</td>
</tr>
</tbody>
</table>

Order Declared Known:
V1, V4, V2, V3, V7
Find MST using Prim’s

Start with $V_1$

<table>
<thead>
<tr>
<th>V</th>
<th>Kwn</th>
<th>Distance</th>
<th>path</th>
</tr>
</thead>
<tbody>
<tr>
<td>v1</td>
<td>Y</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>v2</td>
<td>Y</td>
<td>2</td>
<td>V1</td>
</tr>
<tr>
<td>v3</td>
<td>Y</td>
<td>2</td>
<td>V4</td>
</tr>
<tr>
<td>v4</td>
<td>Y</td>
<td>1</td>
<td>V1</td>
</tr>
<tr>
<td>v5</td>
<td></td>
<td>6</td>
<td>V7</td>
</tr>
<tr>
<td>v6</td>
<td>Y</td>
<td>1</td>
<td>V7</td>
</tr>
<tr>
<td>v7</td>
<td>Y</td>
<td>4</td>
<td>V4</td>
</tr>
</tbody>
</table>

Order Declared Known: V1, V4, V2, V3, V7, V6
Find MST using Prim's

Start with \( V_1 \)

Order Declared Known:
\( V1, V4, V2, V3, V7, V6, V5 \)

<table>
<thead>
<tr>
<th>V</th>
<th>Kwn</th>
<th>Distance</th>
<th>path</th>
</tr>
</thead>
<tbody>
<tr>
<td>v1</td>
<td>Y</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>v2</td>
<td>Y</td>
<td>2</td>
<td>V1</td>
</tr>
<tr>
<td>v3</td>
<td>Y</td>
<td>2</td>
<td>V4</td>
</tr>
<tr>
<td>v4</td>
<td>Y</td>
<td>1</td>
<td>V1</td>
</tr>
<tr>
<td>v5</td>
<td>Y</td>
<td>6</td>
<td>V7</td>
</tr>
<tr>
<td>v6</td>
<td>Y</td>
<td>1</td>
<td>V7</td>
</tr>
<tr>
<td>v7</td>
<td>Y</td>
<td>4</td>
<td>V4</td>
</tr>
</tbody>
</table>
Find MST using Prim’s

Start with $V_1$

<table>
<thead>
<tr>
<th>V</th>
<th>Kwn</th>
<th>Distance</th>
<th>path</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_1$</td>
<td>Y</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$v_2$</td>
<td>Y</td>
<td>2</td>
<td>$V_1$</td>
</tr>
<tr>
<td>$v_3$</td>
<td>Y</td>
<td>2</td>
<td>$V_4$</td>
</tr>
<tr>
<td>$v_4$</td>
<td>Y</td>
<td>1</td>
<td>$V_1$</td>
</tr>
<tr>
<td>$v_5$</td>
<td>Y</td>
<td>6</td>
<td>$V_7$</td>
</tr>
<tr>
<td>$v_6$</td>
<td>Y</td>
<td>1</td>
<td>$V_7$</td>
</tr>
<tr>
<td>$v_7$</td>
<td>Y</td>
<td>4</td>
<td>$V_4$</td>
</tr>
</tbody>
</table>

Order Declared Known: $V_1, V_4, V_2, V_3, V_7, V_6, V_5$

Selected Edges:
Find MST using Prim’s

Start with $V_1$

Order Declared Known:
$V_1, V_4, V_2, V_3, V_7, V_6, V_5$

Selected Edges:
$\{V_2, V_1\}, \{V_3, V_4\}, \{V_4, V_1\}, \{V_5, V_7\}, \{V_6, V_7\}, \{V_7, V_4\}$

<table>
<thead>
<tr>
<th>V</th>
<th>Kwn</th>
<th>Distance</th>
<th>path</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_1$</td>
<td>Y</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$v_2$</td>
<td>Y</td>
<td>2</td>
<td>$V_1$</td>
</tr>
<tr>
<td>$v_3$</td>
<td>Y</td>
<td>2</td>
<td>$V_4$</td>
</tr>
<tr>
<td>$v_4$</td>
<td>Y</td>
<td>1</td>
<td>$V_1$</td>
</tr>
<tr>
<td>$v_5$</td>
<td>Y</td>
<td>6</td>
<td>$V_7$</td>
</tr>
<tr>
<td>$v_6$</td>
<td>Y</td>
<td>1</td>
<td>$V_7$</td>
</tr>
<tr>
<td>$v_7$</td>
<td>Y</td>
<td>4</td>
<td>$V_4$</td>
</tr>
</tbody>
</table>
Prim’s Algorithm Analysis

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

Correctness:
Proof is similar to Dijkstra’s
Kruskal’s MST Algorithm

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

\[ G=(V,E) \]
Kruskal’s Algorithm for MST

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

1. Initialize with
   - empty MST
   - all vertices marked unconnected
   - all edges unmarked

2. While there are still unmarked edges
   a. Pick the lowest cost edge \((u,v)\) and mark it
   b. If \(u\) and \(v\) are not already connected, add \((u,v)\) to the MST and mark \(u\) and \(v\) as connected

Sound familiar?
Data Structures for Kruskal

• Sorted edge list

  \{7,4\} \{2,1\} \{7,5\} \{5,6\} \{5,4\} \{1,6\} \{2,7\} \{2,3\} \{3,4\} \{1,5\}

  0  1  1  2  2  3  3  3  3  4

• Disjoint Union / Find
  – Union(a,b):
    union the disjoint sets named by a and b
  – Find(a):
    returns the name of the set containing a
Kruskal Example

\[
\begin{align*}
\{7,4\} & \quad \{2,1\} & \quad \{7,5\} & \quad \{5,6\} & \quad \{5,4\} & \quad \{1,6\} & \quad \{2,7\} & \quad \{2,3\} & \quad \{3,4\} & \quad \{1,5\} \\
0 & \quad 1 & \quad 1 & \quad 2 & \quad 2 & \quad 3 & \quad 3 & \quad 3 & \quad 3 & \quad 4
\end{align*}
\]
Kruskal Example

\{7,4\} \{2,1\} \{7,5\} \{5,6\} \{5,4\} \{1,6\} \{2,7\} \{2,3\} \{3,4\} \{1,5\}

0 1 1 2 2 3 3 3 3 4
Kruskal Example

\{7,4\} \{2,1\} \{7,5\} \{5,6\} \{5,4\} \{1,6\} \{2,7\} \{2,3\} \{3,4\} \{1,5\}

0 1 1 2 2 3 3 3 3 3 4
Kruskal Example
Kruskal Example

\{7,4\} \{2,1\} \{7,5\} \{5,6\} \{5,4\} \{1,6\} \{2,7\} \{2,3\} \{3,4\} \{1,5\}

0 1 1 2 2 3 3 3 3 4
Kruskal Example

\{7,4\} \{2,1\} \{7,5\} \{5,6\} \{5,4\} \{1,6\} \{2,7\} \{2,3\} \{3,4\} \{1,5\}

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>1</th>
<th>2</th>
<th>2</th>
<th>3</th>
<th>3</th>
<th>3</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
</table>

Diagram: A network with labeled nodes and edges, each edge labeled with a weight.
Kruskal Example

\{7,4\} \{2,1\} \{7,5\} \{5,6\} \{5,4\} \{1,6\} \{2,7\} \{2,3\} \{3,4\} \{1,5\}

0 1 1 2 2 3 3 3 3 4
Kruskal Example

\{7,4\} \{2,1\} \{7,5\} \{5,6\} \{5,4\} \{1,6\} \{2,7\} \{2,3\} \{3,4\} \{1,5\}

0 1 1 2 2 3 3 3 3 4
Kruskal Example

\{7,4\} {2,1} {7,5} {5,6} {5,4} {1,6} {2,7} {2,3} {3,4} {1,5}

0 1 1 2 2 3 3 3 3 4
Kruskal Example

\{7,4\} \{2,1\} \{7,5\} \{5,6\} \{5,4\} \{1,6\} \{2,7\} \{2,3\} \{3,4\} \{1,5\}

0 1 1 2 2 3 3 3 3 4
Example of DU/F

Find(5) = 7
Find(4) = 7

{7,4}  {2,1}  {7,5}  {5,6}  {5,4}  {1,6}  {2,7}  {2,3}  {3,4}  {1,5}
0   1   1   2   2   3   3   3   3   4
Example of DU/F

Find(1) = 1
Find(6) = 7

{7, 4} {2, 1} {7, 5} {5, 6} {5, 4} {1, 6} {2, 7} {2, 3} {3, 4} {1, 5}

0 1 1 2 2 3 3 3 3 3 4
Example of DU/F

Union(1,7)

{7,4} {2,1} {7,5} {5,6} {5,4} {1,6} {2,7} {2,3} {3,4} {1,5}

0 1 1 2 2 3 3 3 4
Kruskal’s Algorithm with DU / F

Sort the edges by increasing cost;
Initialize A to be empty;
for each edge \{i,j\} chosen in increasing order do
  u := Find(i);
  v := Find(j);
  if not(u = v) then
    add \{i,j\} to A;
    Union(u,v);
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);
        }
    }
}
Find MST using Kruskal’s
Find MST using Kruskal’s
Find MST using Kruskal’s
Find MST using Kruskal’s
Find MST using Kruskal’s
Find MST using Kruskal’s
Find MST using Kruskal’s
Find MST using Kruskal’s

Is this MST unique?

Under what condition is an MST unique?
  - Unique edge weights guarantee uniqueness