MST

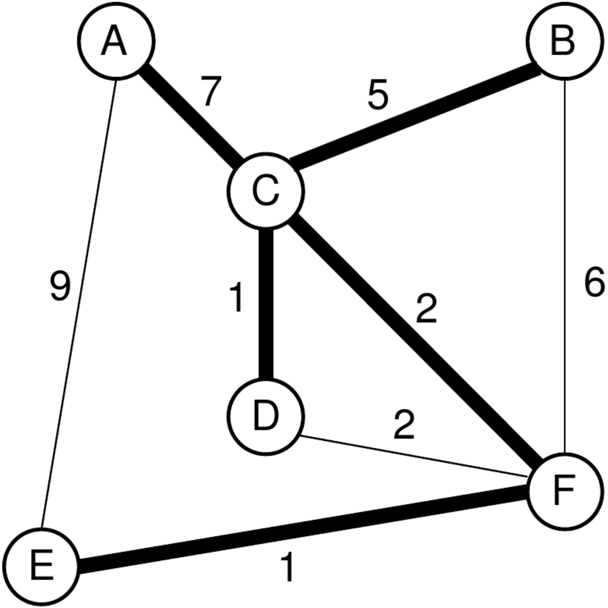
Comp Sci 1575 Data Structures

Outline

Outline

- The lowest cost solution to unifying all your future destinations...

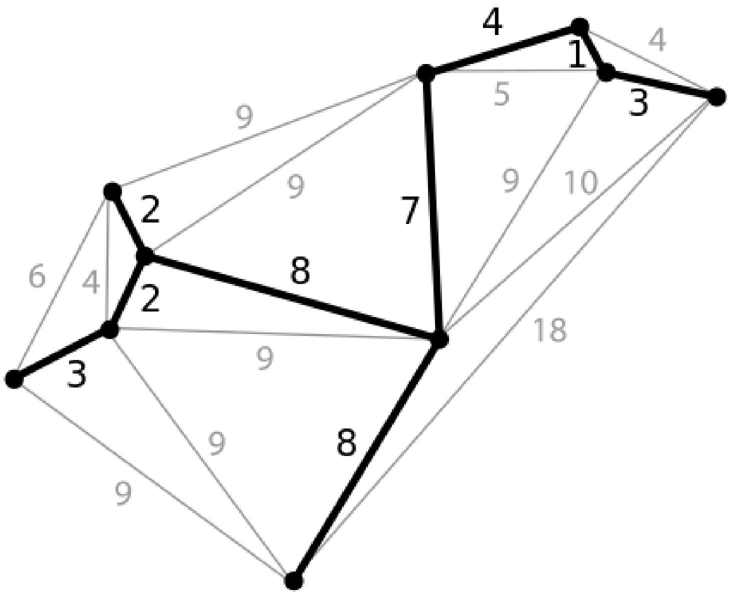
Minimum-cost Spanning Tree



MST

- is a subset of edges of a connected, edge-weighted (un)directed graph that connects all the vertices together, without any cycles and with the minimum possible total edge weight. That is, it is a spanning tree whose sum of edge weights is as small as possible.
- is the graph containing the vertices of G along with the subset of G's edges that
  - (1) has minimum total cost as measured by summing the values for all of the edges in the subset, and
  - (2) keeps the vertices connected.
- MST contains no cycles. If a proposed MST did have a cycle, a cheaper MST could be had by removing any one of the edges in the cycle.
- MST is a free tree with $|V| - 1$ edges.
- **"minimum-cost spanning tree"** comes from the fact that the required set of edges forms a tree, it spans the vertices (i.e., it connects them together), and it has minimum cost.

Minimum-cost Spanning Tree



Outline

1. Introduction
   - Definitions
   - Applications
   - Variations
2. Properties
   - Uniqueness
   - Cut
   - Cycle
3. Algorithms
   - Prim's
     - Example 1
     - Example 2
     - Example 3

Applications

- Direct applications in the design of networks, including computer networks, telecommunications networks, transportation networks, water supply networks, and electrical grids
- Soldering the shortest set of wires needed to connect a set of terminals on a circuit board
- Connecting a set of cities by telephone lines in such a way as to require the least amount of cable
- Cluster analysis
- Image registration and segmentation
- Taxonomy/classification
- Financial market analysis
- Maze generation

Outline

Maximum-cost Spanning Tree

The same algorithm and acronym is also used to describe an analogous structure, the Maximum Spanning Tree.

Outline

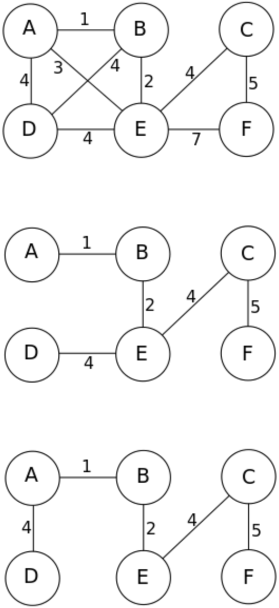
Outline

1 Introduction
- Definitions
- Applications
- Variations

2 Properties
- Uniqueness
- Cut
- Cycle

3 Algorithms
- Prim's
  - Example 1
  - Example 2
  - Example 3

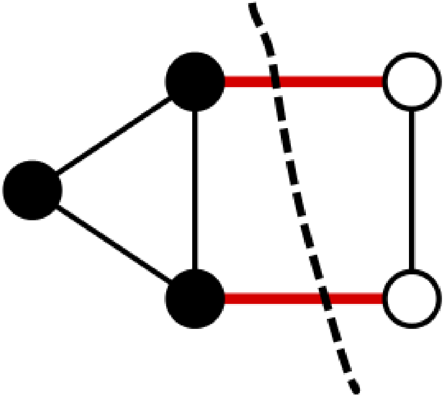
Uniqueness



- If there are n vertices in the graph, then each spanning tree has n - 1 edges.
- There may be several minimum spanning trees of the same weight; in particular, if all the edge weights of a given graph are the same, then every spanning tree of that graph is minimum.
- If each edge has a distinct weight then there will be only one, unique minimum spanning tree.

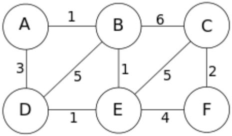
Outline

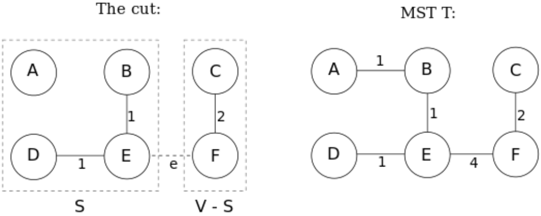
Graph cut



- In graph theory, a cut is a partition of the vertices of a graph into two disjoint subsets.
- Any cut determines a cut-set, the set of edges that have one endpoint in each subset of the partition. These edges are said to cross the cut.

Cut property





For any cut C of a graph, if the weight of an edge e in the cut-set of C is strictly smaller than the weights of all other edges of the cut-set of C, then this edge belongs to all MSTs of the graph.

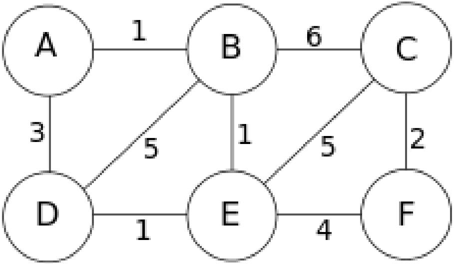
Outline

1 Introduction
- Definitions
- Applications
- Variations

2 Properties
- Uniqueness
- Cut
- Cycle

3 Algorithms
- Prim's
  - Example 1
  - Example 2
  - Example 3

Cycle

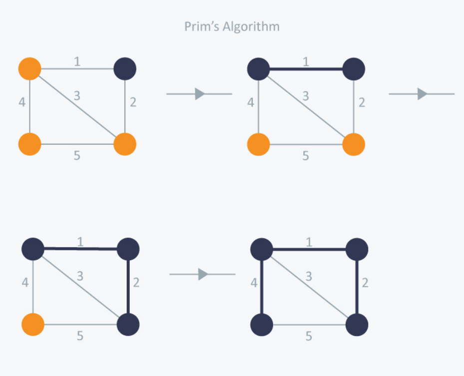


For any cycle C in the graph, if the weight of an edge e of C is larger than the individual weights of all other edges of C, then this edge cannot belong to a MST.

Outline

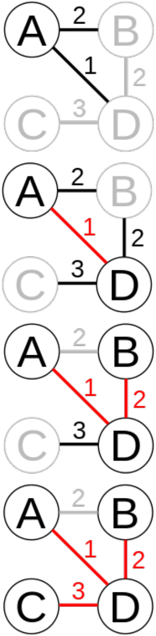
Outline

Prim's algorithm: one method to find a MST



1. Initialize a tree with a single vertex, chosen arbitrarily from the graph.
2. Grow the tree by one edge: of the edges that connect the tree to vertices not yet in the tree, find the minimum-weight edge, and transfer it to the tree.
3. Repeat step 2 (until all vertices are in the tree).

Prim's algorithm

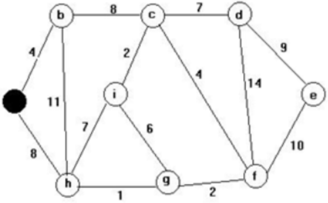


Grow the spanning tree from a starting position.

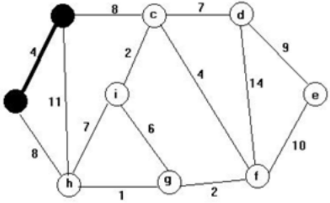
1. Maintain two disjoint sets of vertices. One containing vertices that are in the growing spanning tree and other that are not in the growing spanning tree.
2. Select the cheapest vertex that is connected to the growing spanning tree and is not in the growing spanning tree and add it into the growing spanning tree. This can be done using Priority Queues. Insert the vertices, that are connected to growing spanning tree, into the Priority Queue.
3. Check for cycles. To do that, mark the nodes which have been already selected and insert only those nodes in the Priority Queue that are not marked.

Prim's algorithm: example

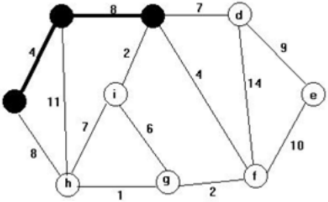
Iteration 0:



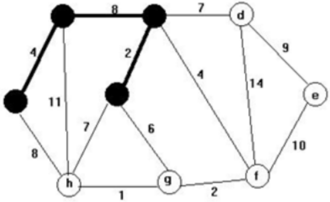
Iteration 1:



Iteration 2:

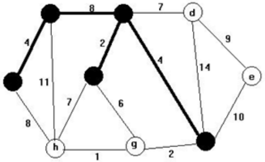


Iteration 3:

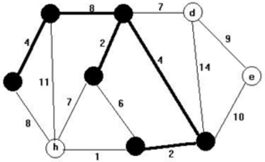


Prim's algorithm: example

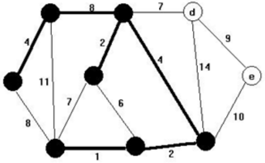
Iteration 4:



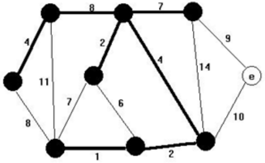
Iteration 5:



Iteration 6:



Iteration 7:



Iteration 8: