Minimum Spanning Tree problem

Data structures

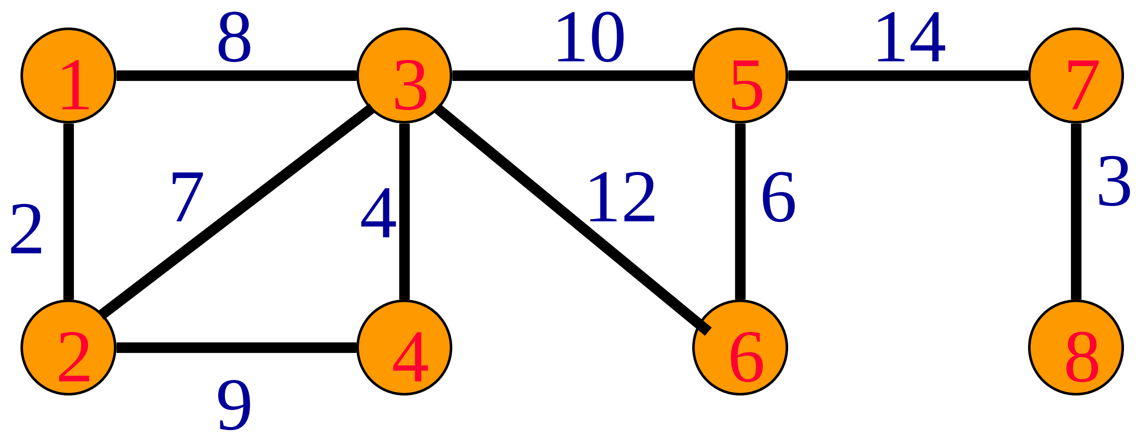
Spring 2017

Minimum-Cost Spanning Tree

- Given a weighted connected undirected graph:
- find a spanning tree that has minimum cost

(cost of spanning tree is sum of edge costs)

Example



- Network has 10 edges.
- Spanning tree has only n - 1 = 7 edges.
- Need to either select 7 edges or discard 3.

Edge Selection Greedy Strategies

- Start with an n-vertex 0-edge forest. Consider edges in ascending order of cost. Select edge if it does not form a cycle together with already selected edges.
  - Kruskal’s method.
- Start with a 1-vertex tree and grow it into an n-vertex tree by repeatedly adding a vertex and an edge. When there is a choice, add a least cost edge.
  - Prim’s method.

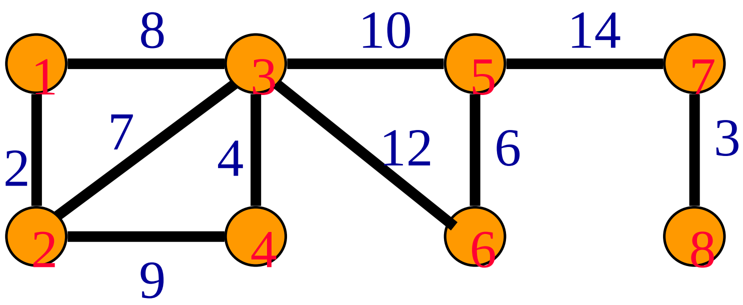
Edge Selection Greedy Strategies

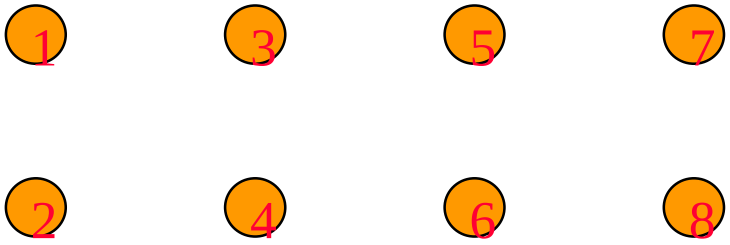
- Start with an n-vertex forest. Each component/tree selects a least cost edge to connect to another component/tree. Eliminate duplicate selections and possible cycles. Repeat until only 1 component/tree is left.
  - Sollin's method.

Edge Rejection Greedy Strategies

- Start with the connected graph. Repeatedly find a cycle and eliminate the highest cost edge on this cycle. Stop when no cycles remain.
- Consider edges in descending order of cost. Eliminate an edge provided this leaves behind a connected graph.

Kruskal's Method

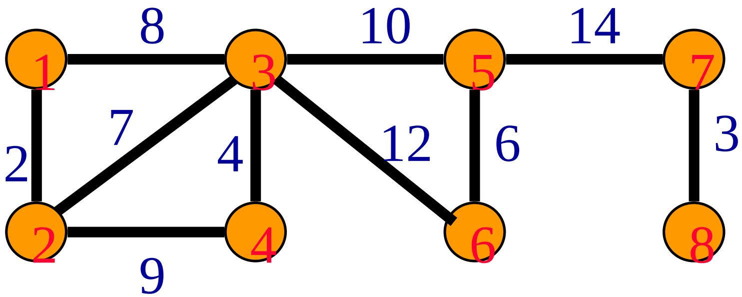


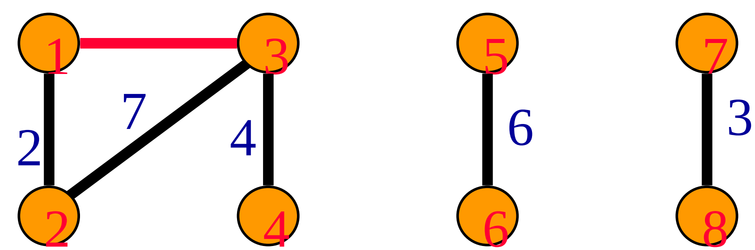


- Start with a forest that has no edges.

- Consider edges in ascending order of cost.
- Edge (1,2) is considered first and added to the forest.

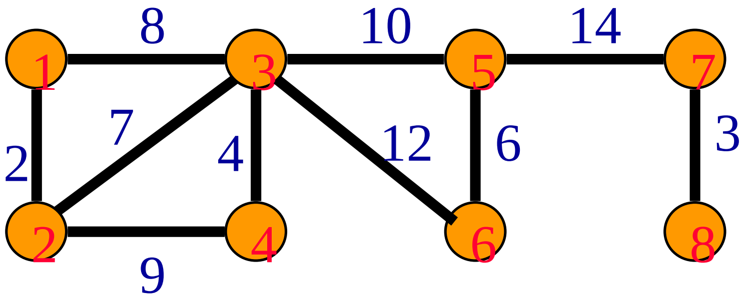
Kruskal's Method

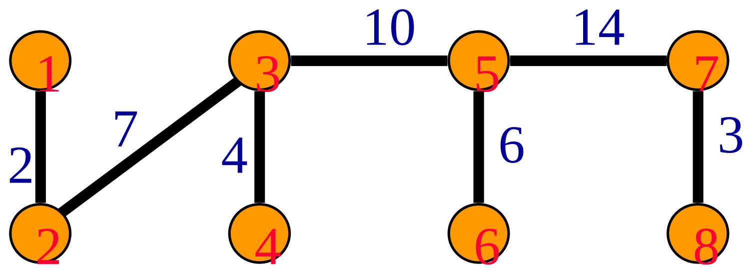




- Edge (7,8) is considered next and added.
- Edge (3,4) is considered next and added.
- Edge (5,6) is considered next and added.
- Edge (2,3) is considered next and added.
- Edge (1,3) is considered next and rejected because it creates a cycle.

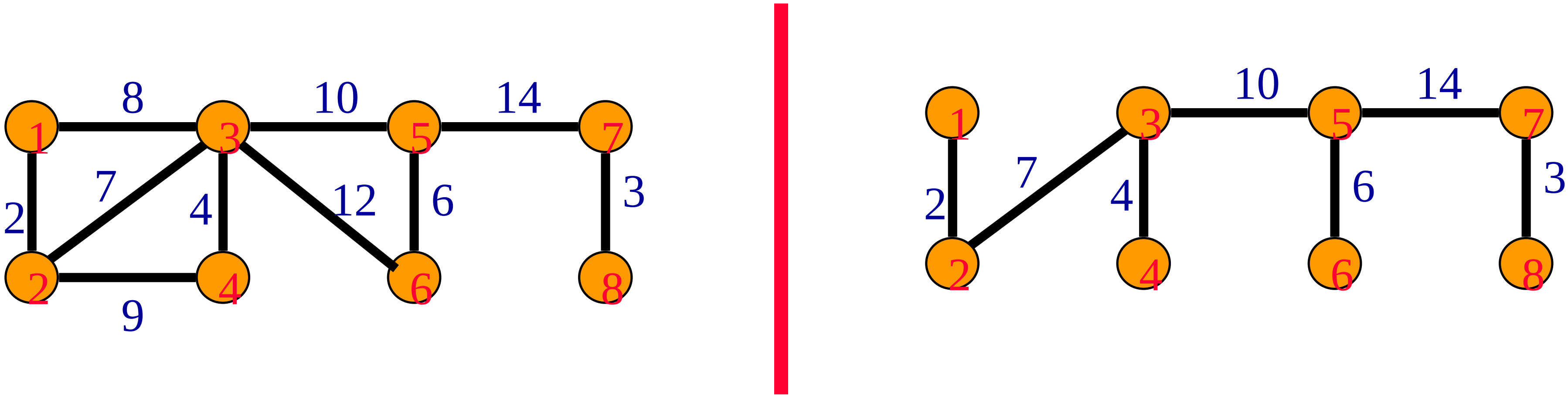
Kruskal's Method





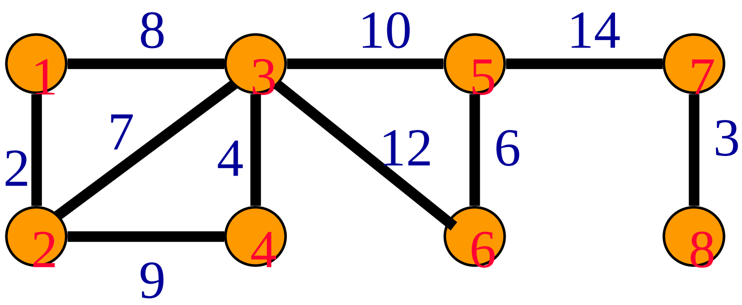
- Edge (2,4) is considered next and rejected because it creates a cycle.
- Edge (3,5) is considered next and added.
- Edge (3,6) is considered next and rejected.
- Edge (5,7) is considered next and added.

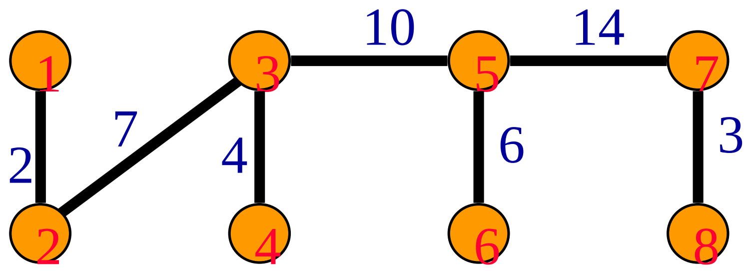
Kruskal's Method



- n - 1 edges have been selected and no cycle formed.
- So we must have a spanning tree.
- Cost is 46.
- Min-cost spanning tree is unique when all edge costs are different.

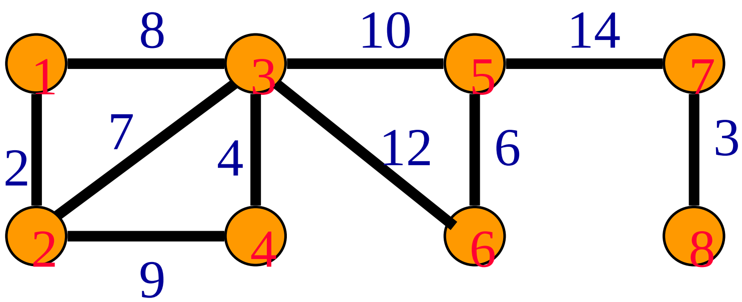
Prim's Method

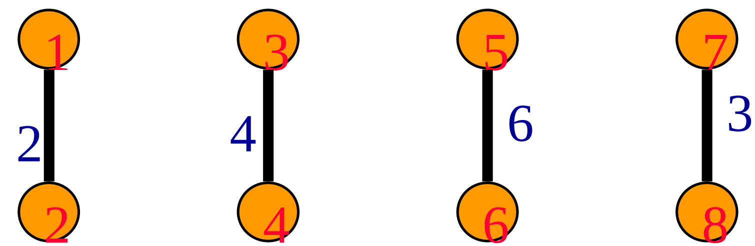




- Start with any single vertex tree.
- Get a 2-vertex tree by adding a cheapest edge.
- Get a 3-vertex tree by adding a cheapest edge.
- Grow the tree one edge at a time until the tree has n - 1 edges (and hence has all n vertices).

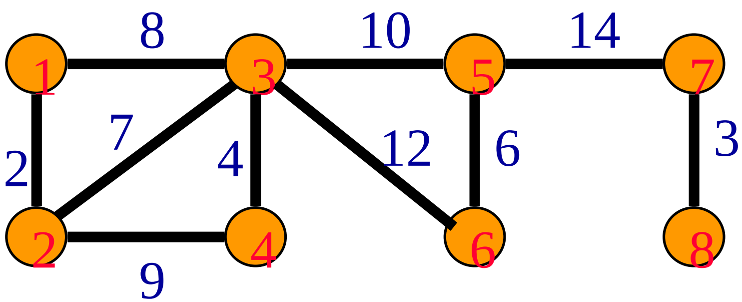
Sollin's Method

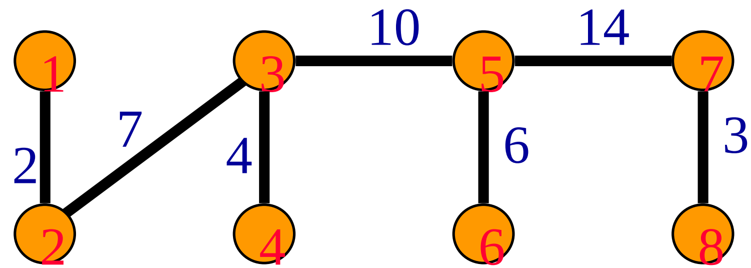




- Start with a forest that has no edges.
- Each component selects a least cost edge with which to connect to another component.
- Duplicate selections are eliminated.
- Cycles are possible when the graph has some edges that have the same cost.

Sollin's Method





- Each component that remains selects a least cost edge with which to connect to another component.
- Beware of duplicate selections and cycles.

Greedy Minimum-Cost Spanning Tree Methods

- Can prove that all the algorithms return a minimum-cost spanning tree.
- Prim's method is fastest.
  - $O(n^2)$ using an implementation similar to that of Dijkstra's shortest-path algorithm.
  - $O(e + n \log n)$ using a Fibonacci heap.
- Kruskal's uses union-find trees to run in $O(n + e \log e)$ time.