Lecture 12
Minimum Spanning Trees

Department of Computer Science
Hofstra University

Lecture Goals

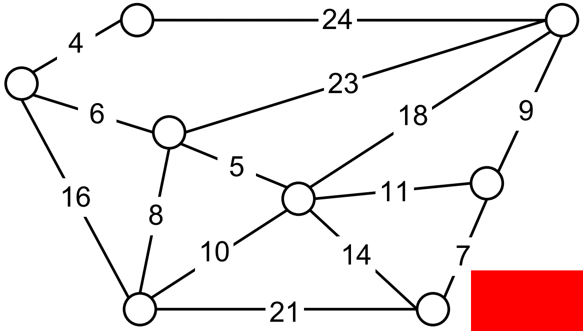
- In this lecture we study the minimum spanning tree problem.
- We consider two classic algorithm for the problem—Kruskal's algorithm and Prim's algorithm.
  - Both are greedy algorithms that are also optimal.

Minimum Spanning Tree (MST)

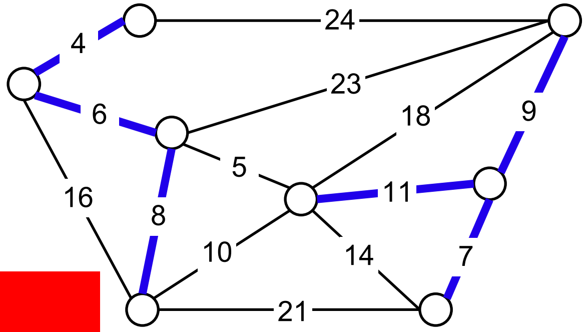
Given. Undirected graph G with positive edge weights (connected).

Def. A spanning tree of G is a subgraph T that is both a tree (connected and acyclic) and spanning (includes all of the vertices).

Goal. Find a min weight spanning tree.

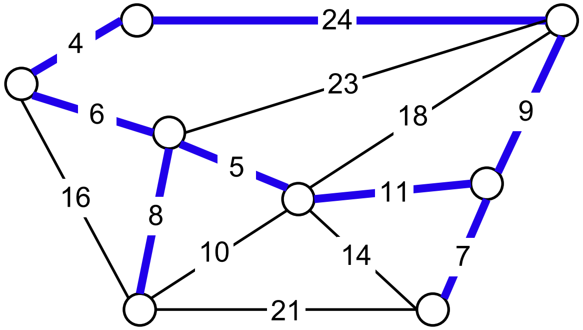


graph G

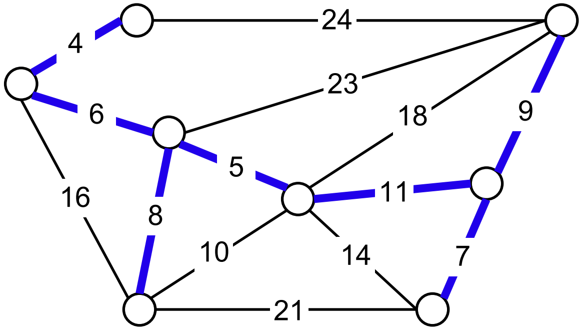


not connected

Brute force.
Try all spanning trees?



not acyclic

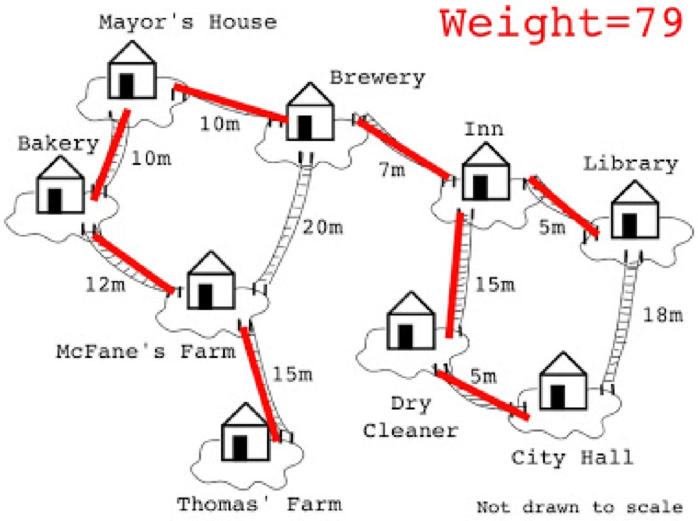


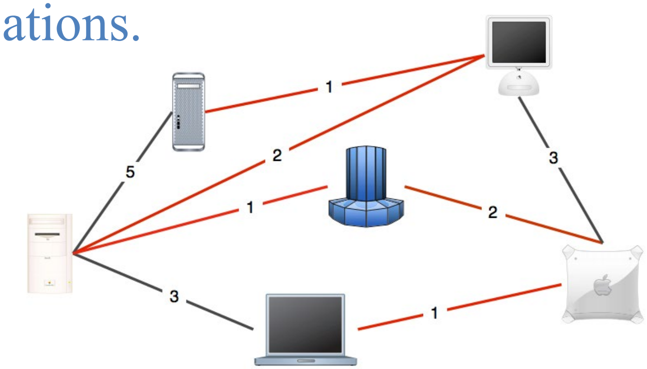
spnning tree T: cost = 50 = 4 + 6 + 8 + 5 + 11 + 9 + 7

MST Applications

- One example would be a telecommunications company trying to lay cable in a new neighborhood. If it is constrained to bury the cable only along certain paths (e.g. roads), then there would be a graph containing the points (e.g. houses) connected by those paths.
- Some of the paths might be more expensive, because they are longer, or require the cable to be buried deeper; these paths would be represented by edges with larger weights.
- A *MST* would be one with the lowest total cost, representing the least expensive path for laying the cable.

- Network design.
- Cluster analysis.
- Indirect applications.



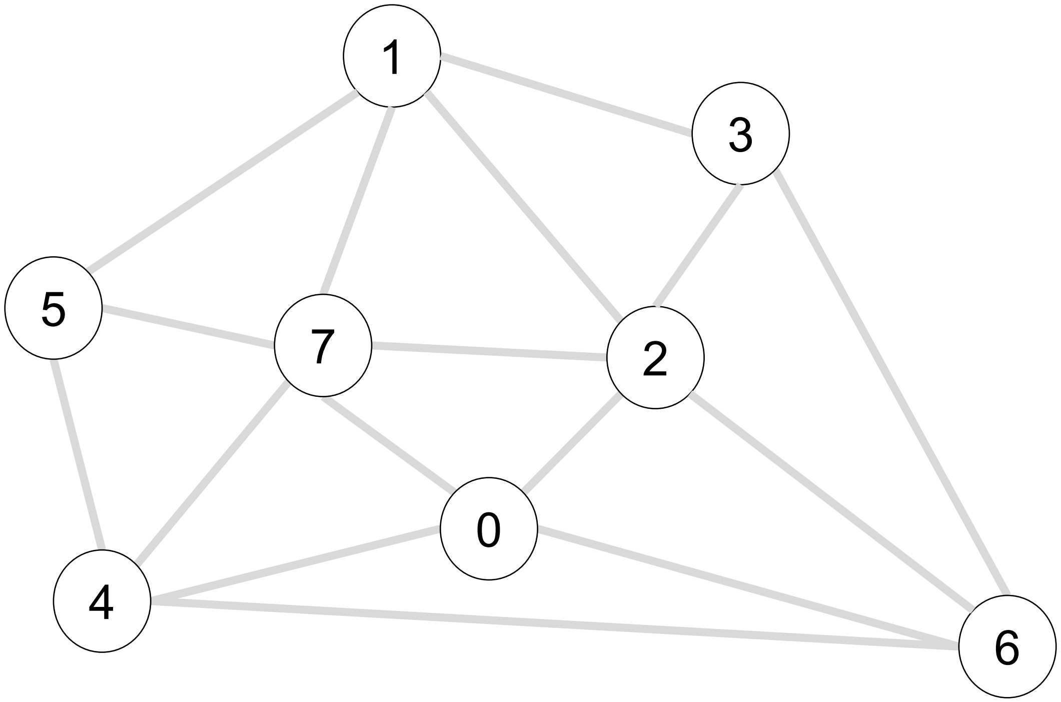


Kruskal’s Algorithm

- Consider edges in ascending order of weight.
- Add next edge to tree T unless doing so would create a cycle.

does not create a cycle

creates a cycle



an edge-weighted graph

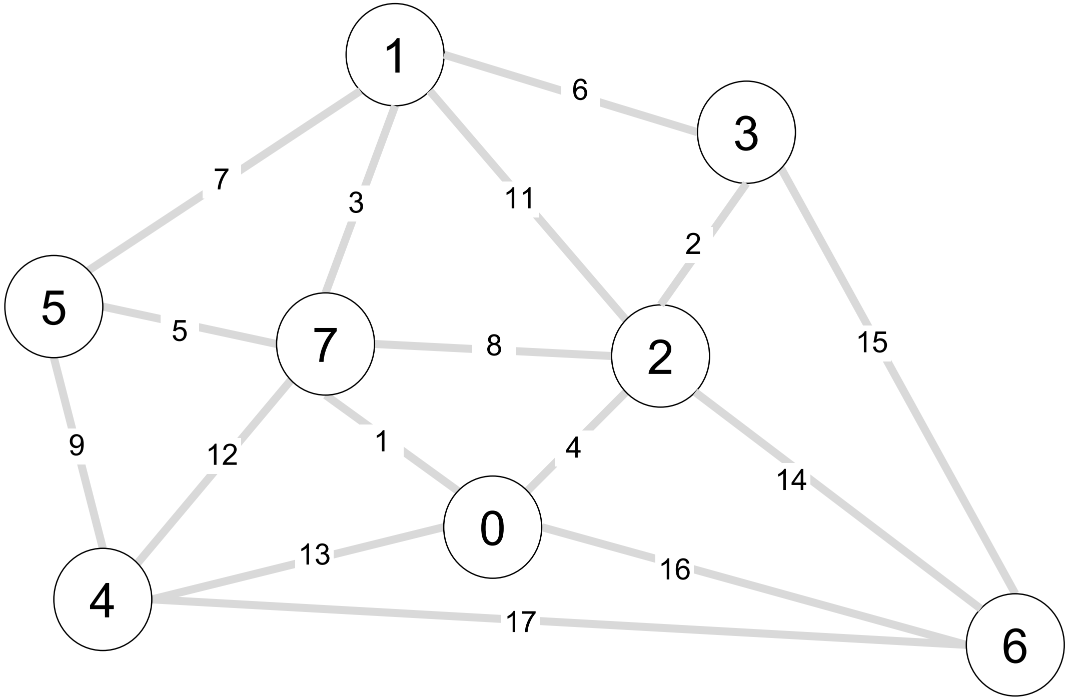
Kruskal's algorithm in 2 minutes
https://www.youtube.com/watch?v=71UQH7Pr9kU

graph edges sorted by weight

0 - 7	1	←
2 - 3	2	←
1 - 7	3	←
0 - 2	4	←
5 - 7	5	←
1 - 3	6	←
1 - 5	7	←
2 - 7	8	←
4 - 5	9	←
1 - 2	10	←
4 - 7	11	←
0 - 4	12	←
2 - 6	13	←
3 - 6	14	←
0 - 6	15	←
4 - 6	16	←

Prim’s Algorithm

- Start with vertex 0 and greedily grow tree T.
- Add to T the min weight edge with exactly one endpoint in T.
- Repeat until V – 1 edges.

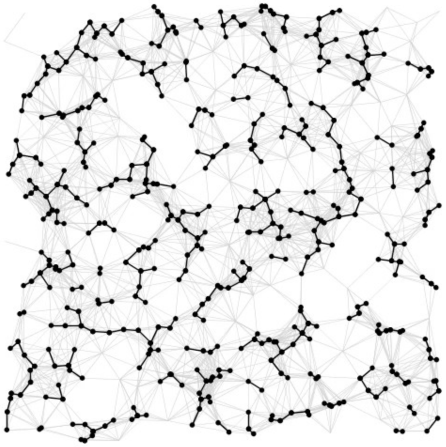
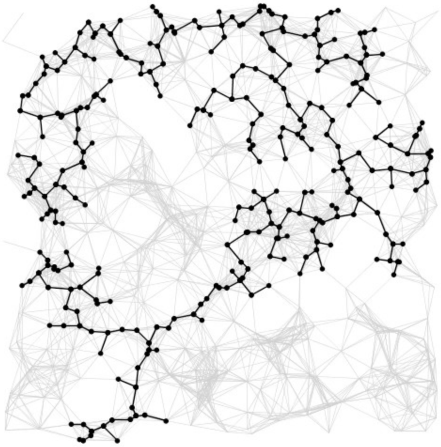


Prim's algorithm in 2 minutes
https://www.youtube.com/watch?v=cplfcGZmX7I

Dijkstra’s Algorithm vs. Prim’s algorithm

- Similarities:
  - Both use a greedy approach.
  - They employ similar data structures, often using a priority queue.
  - The basic structure of both algorithms is very similar, with the main difference being in how they update vertex values
- Dijkstra's finds shortest paths, while Prim's constructs a minimum spanning tree (MST).
  - This is reflected in how they calculate and update vertex values:
  - In Dijkstra's algorithm, choose the closest vertex to the source node (via a directed path), the distance to a vertex is relaxed to sum of the edge weight plus the distance to the previous vertex if it is smaller.
  - In Prim's algorithm, choose the closest vertex to the tree (via an undirected edge), i.e., the vertex with the minimum weight of the edge connecting it to the MST.

Summary

algorithm	visualization	bottleneck	running time
Kruskal		sorting union-find	$E \log V$
Prim		priority queue	$E \log V$

https://www.youtube.com/watch?v=vmWSnkBVvQ0