Lecture 40: Minimum Spanning Trees

CSCI 62
Fall, 2016

Kim Bruce & Peter Mawhorter

Spanning Trees

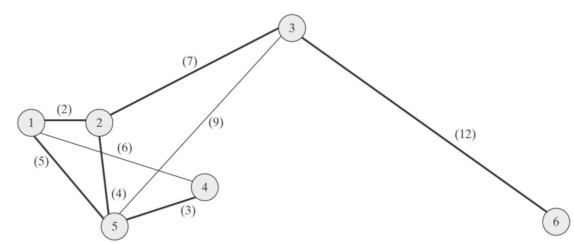
- A spanning tree T of a graph G is a subset of the edges of G such that:
  - T contains no cycles and
  - Every vertex in G is connected to every other vertex using just the edges in T.
- An unconnected graph has no spanning trees.
- A connected graph G will have at least one spanning tree; it may have many.

Minimum Spanning Trees

- A weighted graph is a graph that has a weight associated with each edge.
- If G is a weighted graph, the cost of a tree is the sum of the costs (weights) of its edges.
- A tree T is a minimum spanning tree of G iff:
  - it is a spanning tree and
  - there is no other spanning tree whose cost is lower than that of T.

Don't care about the root!

Minimum Spanning Trees



- Can we find an MST without searching all the possible trees?

Minimum Spanning Trees

- Application:
  - The cheapest way to lay cable that connects a set of points is along a minimum spanning tree that connects those points.
- Many algorithms exist to find minimum spanning trees, most run in O(e log e) time.
- In 1995 Karger, Klein & Tarjan found a linear time randomized algorithm, but there is no known linear time deterministic algorithm

Side-Remarks

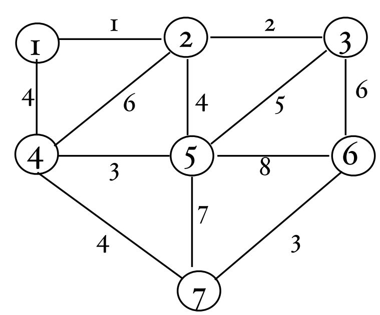
- What is the size of a spanning tree of G if G has n vertices?
- Why must minimum cost spanning tree include least weight edge in graph?

Kruskal's Algorithm

- Create forest F with no edges, using vertices in V
  - Sort the edges in the graph by their weight (smallest to largest)
  - For each edge e in sorted order:
    - if e connects two different trees in F , then add e to F
- Implementation Questions
  - What data structure do you use for the forest?
  - Union-find data structure! (briefly on board)

Sample Graph

(1,2):1
(2,3):2
(4,5):3
(6,7):3
(1,4):4
(2,5):4
(4,7):4
(3,5):5
(2,4):6
(3,6):6
(5,7):7
(5,6):8



Graph Algorithms

- Very important in practice!
- Sophisticated data structures
- Careful analysis of
  - correctness
  - complexity
- CS 140: Algorithms