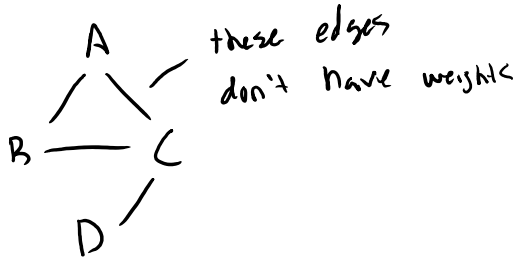
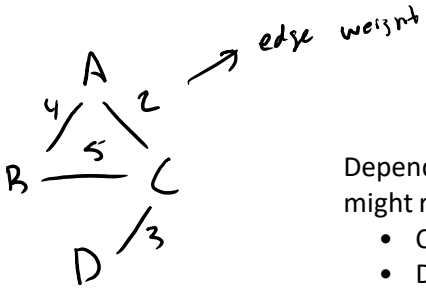
2019-04-23 Minimum Spanning Trees

Tuesday, April 23, 2019 9:03 AM

- Goal of an MST: Reduce a graph to just its components such that the graph remains connected in the most efficient manner
- Efficiency is measured by the sum of each edge's weight
- An aside on edge weight…





Depending on application domain, weights might represent

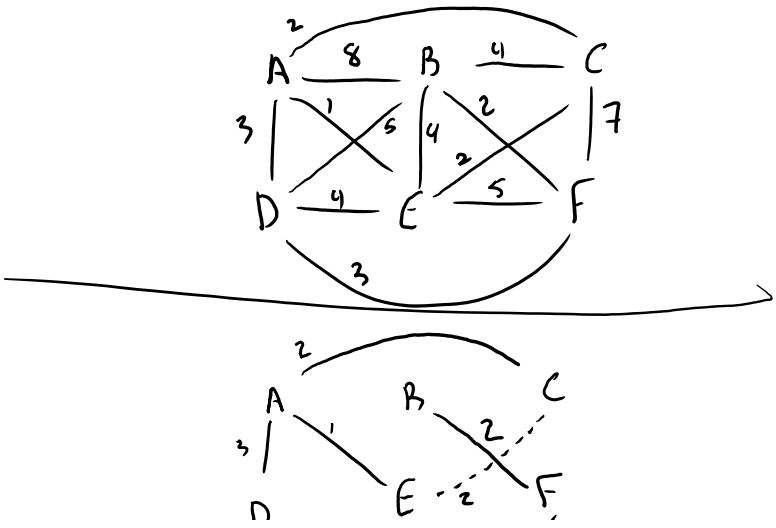
- Cost ($)
- Distance
- Time
- Throughput

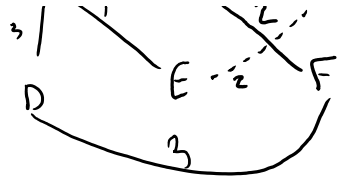
Aside: How do we represent edge weights programmatically?

- Adjacency matrix: store edge weight in cell rather than always 1
- Edge list: store both edge and edge cost (typically as a HT)
  - Key: edge*, Value: edge weight

Back to MSTs…

- Goal of MST is to minimize total edge cost of graph such that all vertices remain somehow connected



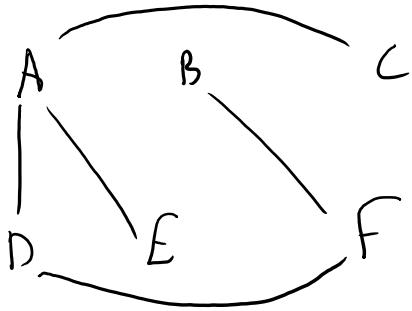


Idea #1: Put all edges into a min PQ

1. AE (1)
2. BF (2)
3. AC (2)
4. EC (2)
5. AD (3)
6. ~~DF (3)~~
7. BC (4)
8. DE (4)
9. BE (4)
10. EF (5)
11. DB (5)
12. CF (7)
13. AB (8)

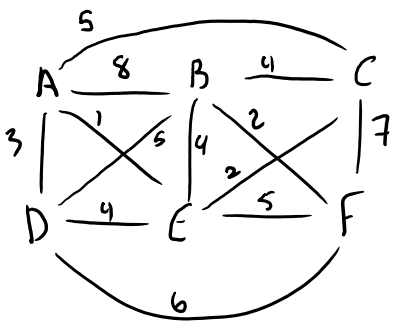
Idea:

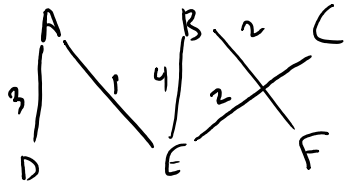
1. Pop edge off
2. Check for redundancy. If none, fill graph



- Problem: How do we determine done-ness?
  - When #edges + 1 = #vertices, done.
- And how do we determine if two vertices are connected?
  - Basic approach: graph search (DFS, BFS) -> SLOW
  - Better approach is to use set operations
    - The disjoint set data structure can perform tests of inclusion and unions in almost O(1) (Log*N which is different and much faster than LogN)
- This algorithm is formally called Kruskal's MST algorithm

Idea #2: Also uses a min PQ





PQ

~~AD (3)~~
AB (8)
AC (5)
~~AE (1)~~
~~EB (4)~~
~~EC (2)~~
ED (4)
EF (5)

CB (4)
CF (7)
CA (5)
DB (5)
DE (4)
DF (6)
BC (4)
BA (8)
BD (5)
~~BF (2)~~

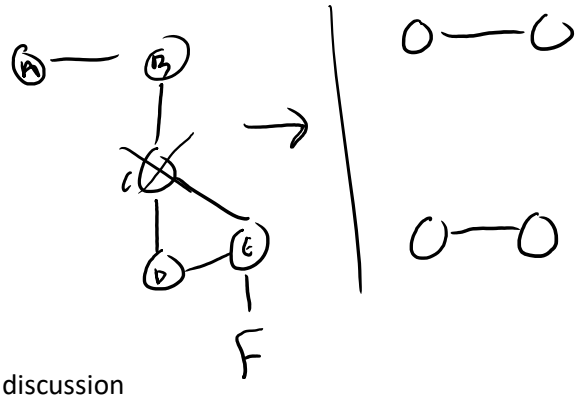
- eIdea: Pic some arbitrary starting location, add all outgoing edges into min PQ.
- While #found edges + 1 < #vertices
  - Pop off top item. If node not

visited previously, take edge. Add its outgoing edges into PQ.

- This algorithm is called Prim's MST

Articulation Point Analysis

- Defined: An articulation point is a "weak point" in the graph such that removing that vertex would create two disconnected graphs.
- Example:



- Examples from class discussion
  - Network router outage
  - Roads
  - Emergency building exits
  - Internet speeds
- In order to programmatically find an articulation point, we must construct a depth-first search tree from our graph
- From above…