Minimal Spanning Tree

- **In an undirected graph, find a minimal spanning tree**
  - Minimal means …
  - Spanning means …
  - Tree means …

- **We can use Prim's Algorithm. Similar to Dijkstra's Algorithm**

  start with initial vertex in MST
  find minimal edge from MST vertex to other vertex
  add minimal edge/vertex to MST
  repeat until each vertex added
Prim's Algorithm

- How do we know this algorithm works?
  - Suppose not, some graph for which algorithm yields non minimal spanning tree, is it not a tree? Not spanning? ...
  - Assume we have such a graph, call it G, now we reason ...
  - Will lead to contradiction (exploit minimal property)

- How efficient is Prim's Algorithm?
  - Depends on implementation, look at Dijkstra and modify
  - What did Dijkstra's algorithm do? Problem solved?
Implementation, priority queue

- Keep track of shortest distance from MST vertices to others, use chosen start vertex
  - Initially all are infinity except start is 0 from itself
  - Choose minimal edge between MST vertices and others
  - This is same as choosing minimal element in PQ

- When choosing, modify distances by adding elements to PQ appropriately
  - When distance < current best modify current best (add new element to PQ)
  - How is this different from Dijkstra's algorithm?
  - How do we construct path?