Kruskal's algorithm
- select edge with smallest weight as accept the edge if it does not cause a cycle
- determining if it causes a cycle: essentially the equivalence class (union/find) problem
- two vertices belong to the same set iff the are connected in the current spanning forest

Prim's algorithm
- grow the tree in successive stages
- in each stage, one node is picked as the root, we add an edge, and thus a vertex is added to the tree
- have a set on vertices in the tree and a set that is not in the tree

Construct MST for this graph using Kruskal's algorithm

Construct MST from v1 for this graph using Prim's algorithm

at each stage, a new vertex to add to the tree is selected by choosing edge (u, v) such that the cost of (u, v) is the smallest among all edges where u is in the tree and v is not
- Build spanning tree starting from v1
- Virtually identical to Dijkstra's, but note that it is for undirected graphs
Connectivity

- Find a path in a graph from one vertex to any other vertex
- Undirected graphs: No separate subgraphs
- Directed graphs: There are some places to which we can get from some directions

Definitions

- Connected undirected graph
- n-connected graphs
- Biconnected graphs
- Cut vertices
- Bridges
- Blocks

Detect Cut Vertices

- Create a tree using the depth-first search algorithm
- Determine Forward edges
- Determine Back edges
- A vertex v if a cut vertex if v has at least one subtree unconnected with any of v's predecessors by back edges

Detect Cut Vertices (cont.)

- A special case: If v is a root with more than one descendant in the tree, then v is a cut vertex

Connectivity in Directed Graphs

- Weakly connected: A corresponding undirected graph is connected.
- Strongly connected: Each pair, there is a path between them in both directions.
- Strongly connected components