equivalent graphs -- must have the same number of vertices and exactly the same set of edges

connected -- given any two vertices in the graph, there is a path from one to the other

complete -- given any two vertices in the graph, there is an edge from one to the other
  · number of edges is \( \frac{v(v - 1)}{2} \), where \( v \) is the number of vertices

tree -- a connected graph satisfying any of the following (equivalent) conditions:
  · has one less edge than vertices ["edges = vertices - 1"]
  · has the smallest possible number of edges for the number of vertices ["minimal"]
  · every edge is needed for the graph to be connected
  · no matter which edge you remove, the graph becomes disconnected ["fragile"]
  · given any two vertices, there is only one path between them ["no alternate routes”]
  · has no circuit
  · anywhere you add an edge, you create a circuit

Euler circuit -- round-trip tour covering every edge exactly once
  · possible only in a connected graph with no vertex of odd degree
  · can find one by never "painting yourself into a corner" (Fleury's algorithm)
  · can find one by finding subcircuits until every edge is used, then “splicing” together

Euler path -- one-way tour covering every edge exactly once
  · possible only in a connected graph with exactly two vertices of odd degree
  · must start at one vertex with odd degree and finish at the other

Hamilton circuit -- round-trip tour visiting every vertex exactly once
  · no general method to tell whether or not a graph has one
  · always possible in a complete graph
  · not possible in a disconnected graph
  · not possible in a tree

Traveling Salesman Problem (TSP) -- Given a complete graph with a cost on each edge,
  find the Hamilton circuit with the lowest total cost.
  · all Hamilton circuits possible since graph is complete
  · only known solution method is "brute force"
  · number of cases grows explosively as size of the graph increases: \( \frac{(N - 1)!}{2} \)
  · called a “hard” problem since the number of cases grows so quickly that it cannot be solved in a reasonable amount of time when graph is fairly large
  · nearest-neighbor and cheapest-link algorithms simple and fast but do not actually solve TSPs since resulting Hamilton circuits not guaranteed to have the lowest total weight
  · advanced algorithms compromise by starting with an easy guess and making adjustments

minimum spanning tree (MST) -- Given a connected graph with a cost on each edge,
  the tree with the lowest total cost that contains all vertices in the given graph
  · easy to construct using Kruskal's algorithm:
Include the least expensive edge. 
Add the next least expensive edge unless it would form a circuit. 
If a circuit would be formed, do not use the most recently added edge. 
As soon as every vertex is included, stop.