1. **Dynamic MSTs**

   Suppose that you already have a minimum spanning tree (MST) in a graph. Now one of the edge weights changes. Give an efficient algorithm to find an MST in the new graph.

2. **Minimum Bottleneck Trees**

   In a graph G, for any pair of vertices u, v, let bottleneck(u, v) be the maximum over all paths p_i from u to v of the minimum-weight edge along p_i. Construct a spanning tree T of G such that for each pair of vertices, their bottleneck in G is the same as their bottleneck in T.

   One way to think about it is to imagine the vertices of the graph as islands, and the edges as bridges. Each bridge has a maximum weight it can support. If a truck is carrying stuff from u to v, how much can the truck carry? We don't care what route the truck takes; the point is that the smallest-weight edge on the route will determine the load.

3. **Eulerian Tours**

   An *Eulerian tour* is a "walk along edges of a graph" (in which successive edges must have a common endpoint) that uses each edge exactly once and ends at the vertex where it starts. A graph is called Eulerian if it has an Eulerian tour.

   Prove that a connected graph is Eulerian iff each vertex has even degree.