1. Kruskal’s algorithm can return different spanning trees for the same input graph $G$, depending on how ties are broken when the edges are initially sorted.

   Show that for every MST $T$ of $G$, there is some way to sort the edges of $G$ in Kruskal’s algorithm so that $T$ will be the MST that is returned.

2. In class, we saw Kruskal’s algorithm, and discussed various Disjoint sets implementations for Kruskal and their running-times. We did not prove Step (iii) of correctness for Kruskal’s Algorithm: that during the execution of KRUSKAL, $(V,F)$ is always contained in some MST of $G$.

   Prove this now.

   It is similar to Step (iii) for PRIM, but not identical.

3. Suppose that all edge weights in a graph $G$ are integers in the range 1 to $|V|$. How fast can you make Kruskal’s algorithm run in this case?

   What if the edge weights are integers in the range from 1 to $C$, for some constant $C$?

   *This is Exercise 23.2-4 of [CLRS]*