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.

*This is Exercise 23.2-1 of [CLRS]*

2. In class on Tuesday 6th Nov, 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 K RUSKAL, \((V,F)\) 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]*

4. Given a point \( p_0 = (x_0, y_0) \), the right horizontal ray from \( p_0 \) is the set of points \( \{ p = (x, y_0) : x \geq x_0 \} \), that is, it is the set of points due right of \( p_0 \). Show how to determine whether a right horizontal ray from a given \( p_0 \) intersects a line segment \( \overline{p_1p_2} \) in \( O(1) \) time, by reducing the problem to that of two line segments intersecting.

*This is Ex. 33.1-6 of [CLRS]*

5. Show that there may be \( \Theta(n^2) \) intersections in a set of \( n \) line segments.

*This is Ex. 33.2-1 of [CLRS]*

Mary Cryan