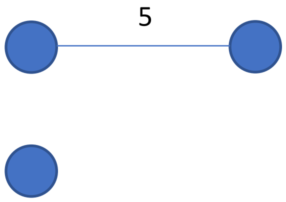
1. Worst Case Algorithmic Asymptotic Complexity:
    a. Kruscal's Algorithm: **O(mlog(m))** – It is actually O(mlogm) + O(mα(m,n)) where 'α' is the inverse ackerman function, which is basically a constant. Therefor, we have O(mlogm) + O(m), which simplifies to O(mlogm).
    b. Prim's Algorithm: **O(mlog(m))** – This is only when using the priority queue implementation, such as I did.
2. The number of edges determines which algorithm you should use. Use Prim's algorithm for a graph with a lot of edges. It can run in O(E + VlogV) time when using a Fibonacci heap implementation (where E = number of edges and V = number of vertices). Kruscal's algorithm, on the other hand, runs in O(ElogV) time.
3. Yes, consider something such as the following:



These two algorithms only work for graphs where all vertices are connected. These algorithms will output the MST for either the top two connected vertices, which will be one edge of length 5, or for the bottom vertex, which will be NULL.