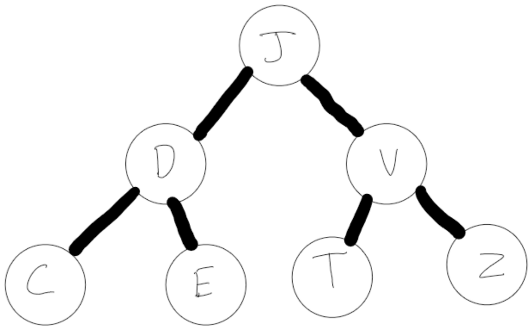
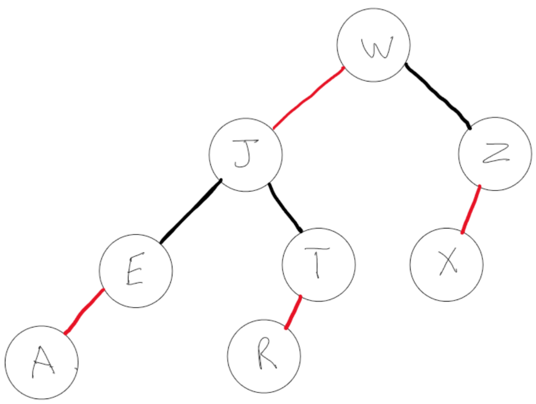
Written Assignment 2

Jonathan Parkes
V00826631

1. a) J, E, T, C, Z, D, V



b) J, E, T, W, X, A, R, Z



c) The number of nodes in any 2-3 tree and its corresponding red-black tree vary from no difference to double the nodes. In the case of a 2-3 tree with inputs to make the max number of nodes the nodes of its corresponding red-black tree will be the exact same where as in the case where the inputs make a 2-3 tree with minimum nodes the corresponding red-black tree will have double the nodes. This is because for any 2 number node in the 2-3 tree there are 2 nodes in the equivalent red-black tree. Because of this when nodes are maximized in a 2-3 tree its red-black tree equivalent is balanced and a 2-3 tree with minimum nodes will have a corresponding unbalanced red-black tree.

2. The algorithm does create a minimum spanning tree. In the algorithm there are 2 parts, when the number of edges are even it performs the Kruskal part, and when the number of edges are odd it does the Prim part.

In the Kruskal part of the algorithm adds the lowest weight edge into the subgraph that does not create cycle. This will always add a new edge and because it is the lowest weight edge not already in the sub graph the only way it can stop a mst from being created is to create a cycle but that cannot happen as it is specified in the algorithm that it will not happen. Therefore this part of the algorithm will always do a proper step to make a mst.

In the prim part of the algorithm we can use the cut property with the any random subgraph. For whatever subgraph was chosen if there is a cut where no edges of the subset cross the cut, the edge of the lowest weight that crosses the cut will be added to the edges of the subtree. By doing this it will always make a mst.

3. For any weighted graph if each edge's weight is unique than there can only be one mst of the graph, meaning it will have a unique mst. If we look at a algorithm such as kruskal's algorithm it adds the lowest weight edge to the mst edges if it doesn't create a cycle, and if there is a tie it chooses one arbitrarily. Therefore the only way a kruskal's algorithm can create multiple different mst's from a single weighted graph is if the graph does not have all unique edges.

Bonus: For Kruskal's algorithm it adds the edge of the lowest weight that does not create a cycle, however if an edge is a tie it chooses one arbitrarily. Therefore to make a mst it doesn't require a weighted graph with unique weights because if there is a tie it chooses one arbitrarily that will never create a cycle meaning it will make a mst.

4.For connected we check if 2 nodes are connected, this results in 2 outcomes either they are connected (in the same tree) or not connected (in different trees). In the case of the 2 nodes being connected each node is at depth at most log(n), therefore it will take at most O(2log(n)) to determine they are connected. This is the same in the case of them being unconnected resulting in O(2log(n)). Therefore at worst case it takes O(2long(n)) = O(log(n)).