**Q No 2 a:**

True. Any linear transformation of all weights maintains all relative path lengths, and thus shortest paths will continue to be shortest paths, and more generally all paths will have the same relative ordering. One simple way of thinking about this is unit conversions between kilometers and miles. For example, if delta is 3, previous path length now becomes (4+3)+(3+3) = 13 by there is another direct edge from a to t i.e. the edge with cost 9 which now became 12 after adding 3, which is now the new shortest path.

**Q No 2 b:**

True. Because if we increase the weight of every edge of G by a positive value δ > 0, then all the edges have the same weight, and minimum spanning of tree remains the same.

**Q No 2 c**

In this case, the given MST T is still a MST. To see this, consider a run of Kruskal’s algorithm which produced T. All of the same decisions would be made when e has a higher weight, and so the same tree will be produced.

**Q No 2 d**

In this case, the given MST T is still a MST because the edges are (n-1) , n stands for the vertices in the graph, then the minimum spanning tree will remain the MST so that we have to only include all the vertices in the tree.

**Algorithm:**

Step 0

Pick any vertex as a starting vertex. (Call it S). Mark it with any given color, say blue or red.

Step 1

Find the nearest neighbor of S (call it P1). Mark both P1 and the edge SP1 blue. The cheapest unmarked/ (uncolored blue) edge in the graph that doesn't close a color blue circuit. Mark it and the edge connecting the vertex to the blue sub graph in blue.

Step 2

Find the nearest uncolored blue neighbor to the blue sub graph (i.e., the closest vertex to any blue vertex). Mark it and the edge connecting the vertex to the blue sub graph in blue.

Step 3

Repeat Step 2 until all vertices are marked blue and few remains red. The blue sub graph is a minimum spanning tree.