CS146 Homework 5

Problem 1: Coded

Problem 2: For the first case if we see that the edges are not in the MST then we need not do anything. (base case). Now for the second case, if the deleted edge was in the MST, then after deletion we get two connected components (sub Minimum Spanning Trees) and to find the new MST, we have to find an edge with minimum weight between two components. A simple implementation would be a simple pass through the edges if the starting and target lies in the set of vertices. To be precise, we can add the minimum weight edge with one vertex for every component. (This gives a time complexity of O(E)).

Problem 3: If a new edge is added then we can find the MST by first determining path between the new edges in the original MST then if it is the maximum edge which is greater than the new edge then we can replace it with the new edge.

Problem 4: As we know in a Dijkstra algorithm, a vertex is marked as visited or closed in turn which means that the algorithm has found the shortest path. The algorithm can find the shortest distance between a node and any other node by continuously calculating the shortest distance from the starting point but if we see that it fails with negative edges. To illustrate let us see an example of three nodes (A, B, C) (A,B, 1), (A, C, 2) , (B, C, -1)) then the algorithm will fail to find A->B->C. The algorithm assumes that any node starting from the a node will lead to greater distance and the algorithm finds the shortest path but this is not true for negative edges.

So for the algorithm it knows the condition that all paths are longer than the root (starting node) and since the algorithm calculates the distances by iterating it is certain that a shorter path than our starting node cannot be found and on the other hand if we have negative edges then the condition is invalid hence the algorithm does not work for negative edges.