**Question One:**

a) revise bellman ford algorithm
b) discuss correctness for handling negative weight cycles
c) run bellman ford for the 5 vertex graph provided

a) The Bellman-ford algorithm functions by considering every path of length k from a single source node; in each successive iteration we consider k+1 and then k+2 edge paths until such time as we have considered every path up to a path length equal to the number of vertices in the graph. For clarity that's the number of edges included as opposed to the cost of that path.

   For any particular iteration, we consider if we can reach some vertex v from u by accessing some new vertex k in between; should that new path be cheaper, we update the existing path to use vertex k.

b) As for why bellman ford is correct when it determines there to be a negative weight cycle in the graph. Once the algorithm proper has terminated, we will have considered every path of length up to and including each and every vertex. The cycle detection part suggests that if we can (after including all paths up to v edges) find a cheaper way of getting to a particular vertex we must have a cycle. The reason this is correct is that if we have included every vertex in our path thus far and we can find a cheaper way of reaching a vertex then we are revisiting a previously found vertex later with less cost after travelling through the entire graph once; this suggests a negative weight cycle as otherwise we would expect the total to be no less than the current cost. having more than v vertices in a path means we can traverse through a negative weight cycle (as that will always be preferenced over other paths as it will always be less) and considering paths beyond the number of vertices includes negative weight cycles where the whole graph is one huge cycle (and of course having a cycle larger than one including every vertex is impossible by definition)

c)

| | i0 | i1 | i2 | i3 | i4 |
|---|---|---|---|---|---|
| s | 0 | 0 | 0 | 0 | 0 |
| u | inf | 5 | 4 | 4 | 4 |
| v | inf | 8 | 8 | 8 | 6 |
| w | inf | inf | 8 | 7 | 7 |
| x | inf | inf | inf | 7 | 6 |

Taking one more iteration we see we can reach u in 2, this is cheaper than the current estimate for u and hence we have a negative weight cycle making the problem of shortest path undefined.

**Question Two:**

See Week 09 lecture slides.

**Question Three:**

See Week 10 lecture slides.

**Question Four:**