Hints to the Solutions: Cpt S 450 Homework #8

1. For each node, I creat an array of k numbers, to indicate the weights of the first k shortest paths from the initial to the node. For the inital node, initially, the array is 0 at the first entry and infinity at all other entries, and for other nodes, initially, the numbers are all infinity.

The algorithm runs in a number of phases. Each phase is to update the arrays stored at all the nodes. Suppose that, before the phse is run, the arrays are W_N, for each node N. We use W'_N to denote the array after the phase: For all nodes M with $M \to N$ as an edge in G, we compute the multiset of numbers: $W_M[i] + Weight(M \to N)$, $0 \leq i \leq k-1$; also, put the numbers in the array W_N stored at the node N into this multiset. Select the first k smallest elements from the multiset and set these elements as the content of W'_N.

After a number of phases when the W'_N's do not change anymore, reverse traverse the graph from final back to the initial and obtain the k shortest paths, making use of the numbers stored in the arrays.

How many phases needed? kn, where n is the number of nodes in G. The algorithm can be improved with further labels on the nodes to indicate whether its is finished (so no updates needed).

2. Many ways to solve this. A straight forward idea is to use the original shortest path algorithm, but at the final state of path retrieval, to avoid the paths with red being followed by yellow.

3. Ideas: build a bag that contains all the color sequences from initial to final; i.e., we have an automaton now. Then, create a forward counting (for number of paths) algorithm from the automaton's initial state (assuming that the automataon does not have a loop – otherwise, the number of color sequences is ininity): for each node, the count of the node is simply the summation of all the counts stored in its parents – the initial stores 1.

4. Easy. Take logarithm.