CS 473: Algorithms, Fall 2009
HBS 4

Problem 1. [Quick Fix]
Your friend suggests that the easiest algorithm for finding shortest paths in a directed graph with negative-weighted edges is to make all the weights positive by adding a sufficiently large constant to each weight and then running Dijkstra's algorithm. Give an example that you can show your friend to prove that his or her method is incorrect.

Problem 2. [Limited Shortest Paths]
We are given a directed graph in which the shortest path between any two vertices u and v is guaranteed to have at most k edges. Give an algorithm that finds the shortest path between two vertices u and v in $O(kE)$ time. Remember, edges can have negative weights.

Problem 3. [Almost Positive]
We are given a directed graph $G = (V, E)$ with potentially negative edge lengths. Your friend ran Dijkstra's algorithm and came up with a shortest path tree T for distances from a node s. You realize that Dijkstra's algorithm may not output distances correctly when a graph has negative edge lengths. However, before you run the more expensive Bellman-Ford algorithm, you wish to check whether T is a correct shortest path tree or not. Describe an $O(m + n)$ time algorithm to do this check. Don't forget to prove that your algorithm is correct!