Problem Set 8

Assigned: July 13
Due: July 20

Problem 1

Consider the following modification of the shortest-paths problem. Edges in the graph are colored red and blue. A path is *valid* if it starts with a red edge, and then alternates colors; that is, it goes red, blue, red, blue ...

Modify the Floyd-Warshall algorithm so as to return the optimal valid path for all pairs of vertices. Assume that the input is a pair of matrices: Red[I,J] is the cost of the edge from I to J if that edge is red, and ∞ otherwise. Blue[I,J] is the cost of the edge from I to J if that edge is blue, and ∞ otherwise.

Problem 2

(CLRS 16.3-5) Give an $O(n^2)$ algorithm to find the longest monotonically increasing subsequence of a sequence of n numbers. For example, in the sequence 10,4,5,11,2,7,4,3,9, the longest increasing subsequence is 4,5,7,9.

Problem 3

A. Give an $O(n^2)$ algorithm that, given a sequence, finds the longest subsequence that first increases then decreases. For instance, in the sequence 10,4,5,11,2,7,4,3,9 the longest such subsequence is 4,5,11,7,4,3. Hint: As with the variants of Floyd-Warshall that we have been discussing (such as problem 1), you want to maintain a number of different arrays corresponding to different categories of subsequences between indices I and J.

B. Consider the modification of (A) which requires that you find the longest subsequence that first increases, then decreases, and does not repeat any values. For instance, given the input sequence, 10,4,5,11,2,7,4,3,9 the subsequence 4,5,11,7,4,3 would be disallowed because the value 4 is repeated. The longest valid subsequence would 4,5,11,7,3. Explain why the kind of dynamic programming technique used in (A) *cannot* be adapted to solve this problem.