Practice Assignment 2

Due: Tuesday, Jan 17th at 2PM to canvas

To get credit, each student must submit their own solutions (which need not be typeset) to TEACH by the due date above – no late submissions are allowed. Note that while you must each submit solutions individually, you are free to work through the problem sets in groups. Solutions will be posted shortly after class and may be discussed in class. These will be not be graded on the basis of *completion* alone, not *correctness*.

1. Give an $O(nK)$ dynamic programming algorithm for the following task:

   Input: A list of n positive integers $a_1, a_2, \ldots, a_n$ and a positive integer K.
   Question: Does some subset of the a_i's add up to K? (You can use each a_i at most once.)

   You should:

   (a) Define the dynamic programming table.
   (b) Give a recursive formula for an entry in the dynamic programming table.
   (c) Describe in words how to fill the dynamic programming table.
   (d) Give pseudocode for the final algorithm.
   (e) Give the running time of your algorithm.

2. Yuckdonald's is considering opening a series of restaurants along Quaint Valley Highway (QVH). The n possible locations are along a straight line, and the distances of these locations from the start of QVH are, in miles and in increasing order, $m_1, m_2, \ldots, m_n$. The constraints are as follows:

   - At each location, Yuckdonald's may open at most one restaurant. The expected profit from opening a restaurant at location i is p_i , where $p_i > 0$ and $i = 1, 2, \ldots, n$.
   - Any two restaurants should be at least k miles apart, where k is a positive integer.

   Give an efficient algorithm to compute the maximum expected total profit subject to the given constraints. You should:

   (a) Define the dynamic programming table.
   (b) Give a recursive formula for an entry in the dynamic programming table.
   (c) Describe in words how to fill the dynamic programming table.
   (d) Give pseudocode for the final algorithm.
   (e) Give the running time of your algorithm.

3. A subsequence is palindromic if it is the same whether read left to right or right to left. For instance, the sequence

   $$A, C, G, T, G, T, C, A, A, A, A, T, C, G$$

   has many palindromic subsequences, including A, C, G, C, A and A, A, A, A (on the other hand, the subsequence A, C, T is not palindromic). Devise an algorithm that takes a sequence $x[1 \ldots n]$ and returns the (length of the) longest palindromic subsequence. Its running time should be $O(n^2)$. You should:

   (a) Define the dynamic programming table.
   (b) Give a recursive formula for an entry in the dynamic programming table.
   (c) Describe in words how to fill the dynamic programming table.
   (d) Give pseudocode for the final algorithm.
   (e) Give the running time of your algorithm.