CS 344 HW 3

Fall 2019

1. (DPV 6.2) You are going on a long trip. You start on the road at mile post 0. Along the way there are n hotels, at mile posts $a_1 < a_2 < \cdots < a_n$, where each a_i is measured from the starting point. The only places you are allowed to stop are at these hotels, but you can choose which of the hotels you stop at. You must stop at the final hotel (at distance a_n), which is your destination.

   You'd ideally like to travel 200 miles a day, but this may not be possible (depending on the spacing of the hotels). If you travel x miles during a day, the penalty for that day is $(200-x)^2$. You want to plan your trip so as to minimize the total penalty – that is, the sum, over all travel days, of the daily penalties.

   Give an efficient algorithm that determines the optimal sequence of hotels at which to stop.

2. (DPV 6.3) 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, $m_1, m_2, ..., m_n$. 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 $i = 1, 2, ..., n$.
   - 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

3. (DPV 6.7) A subsequence is palindromic if it is the same whether read left to right or right to left. For instance, the sequence

   $$\text{A, C, G, T, G, T, C, A, A, A, A, T, C, G}$$

   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 $x[1...n]$ and returns the (length of the) longest palindromic subsequence. Its running time should be $O(n^2)$.

4. (DPV 6.19) Given an unlimited supply of coins of denominations $x_1, x_2, ..., x_n$, we wish to make change for a value v using at most k coins; that is, we wish to find a set of $\leq k$ coins whose total value is v. This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient dynamic-programming algorithm for the following problem.

   Input: $x_1, ..., x_n; k; v$.

   Question: Is it possible to make change for v using at most k coins, of denominations $x_1, ..., x_n$?