CIS 315, Intermediate Algorithms
Winter 2016

Assignment 4

due Monday, February 8, 2016

1. Illustrate the Floyd-Warshall algorithm on the graph described by the following weight matrix:

\[
W = \begin{pmatrix}
0 & \infty & 8 & \infty & \infty \\
6 & 0 & \infty & \infty & 7 \\
\infty & \infty & 0 & \infty & 5 \\
\infty & 3 & \infty & 0 & \infty \\
\infty & \infty & \infty & 4 & 0 \\
\end{pmatrix}
\]

[6 points]

2. exercise R-7.5, p 376 (from GT). More specifically, show that \(O(n^2)\) space can be achieved by dropping the superscripts in algorithm 7.11 (p 355), and the distance matrices \(D^k\) can be computed in place using a single matrix \(D\). The next to last line becomes

\[
D[i, j] = \min\{D[i, j], D[i, k] + D[k, j]\}
\]

[5 points]

The first two steps of the development of a dynamic programming algorithm for a problem are

**step 1** describe the structure of the subproblem

**step 2** find a recurrence for the optimal value of the subproblem in terms of smaller subproblems

Perform just these two steps for the two problems listed below. Do not write (pseudo) code - just the subproblem and recurrence structure.

3. (exercise 6.2 from the DPV text) 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. Perform the two steps above to start the process of determining the minimum possible penalty. [8 points]
4. Imagine a company running a small cloud server which in some months maintains an operation in Siletz (code SIL) and in others in Tillamook (code TIL), moving back and forth between these two cities depending on the local rent and electrical costs (they can only afford to have one office operating at a time). This company wants to have the cheapest possible location plan - the two cities have different operating costs and these costs can change from month to month.

We are given $M$, a fixed cost of moving between the two cities, and lists $S = (s_1, \ldots, s_n)$ and $T = (t_1, \ldots, t_n)$. Here $s_i$ is the cost of operating out of Siletz in month $i$, and $t_i$ is the cost of being in Tillamook that month. Suppose that $M = 10$, $S = (1, 3, 20, 30)$, and $T = (50, 20, 2, 4)$. If the location plan is (SIL, SIL, TIL, TIL), its cost will be $1 + 3 + 10 + 2 + 4 = 20$. On the other hand, the cost of the plan (TIL, TIL, SIL, TIL) is $50 + 20 + 10 + 20 + 10 + 4 = 114$.

The goal here is to (start to) devise a dynamic programming algorithm which, given $M$, $S$, and $T$, determines the cost of the optimal plan. The plan can start in either city, and end in either city. [8 points]

Total: 27 points