1. Illustrate the Floyd-Warshall algorithm by showing each $D^{(k)}$ on the graph described by the following weight matrix:

$$ W = \begin{pmatrix} 0 & \infty & \infty & \infty & -1 & \infty \\ 1 & 0 & \infty & 2 & \infty & \infty \\ \infty & 2 & 0 & \infty & \infty & -8 \\ -4 & \infty & \infty & 0 & 3 & \infty \\ \infty & 7 & \infty & \infty & 0 & \infty \\ \infty & 5 & 10 & \infty & \infty & 0 \end{pmatrix} $$

[8 points]

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

**step 1** define 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 three problems listed below. Do not write (pseudo) code - just the subproblem and recurrence structure.

2. Your company is hired by the WA state highway agency to place warning signs along a dangerous road. On that road are $n$ locations at which you may place a sign, at mile posts $m_1 < m_2 < \cdots < m_n$, where each $m_i$ is measured from the starting point $m_1 = 0$. The only places you are allowed to place a sign are at one of the given mileposts. In addition, you must place one at locations $m_1$ and $m_n$. (If $n = 1$ then you just place one at $m_1$.)

The requirement is to place one every 50 miles, but this may not be possible (depending on the spacing of the mileposts). If you place two consecutive signs $x$ miles apart, the penalty for that placement is $(50 - x)^2$ which will be deducted from your payment. You want to arrange a placement so as to minimize the total penalty - that is, the sum, over all locations, of the penalties. Perform the two steps above to start the process of determining the minimum possible penalty. [8 points]

3. There is a small laboratory specializing in genetic sequencing of viruses which in some months maintains its facility in Sixes, OR (code S) and in others in Takilma, OR (code T), and moves back and forth between these two cities (they can only afford to have one location 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 $M$, the fixed cost of moving all the equipment and employees over the Siskiyou Mountains from one city to the other, 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 Sixes in month $i$, and $t_i$ is the cost of being in Takilma that month. Suppose that $M = 10$, $S = (1, 3, 20, 30)$, and $T = (50, 20, 2, 4)$. If the location plan is $(S, S, T, T)$, its cost will be $1 + 3 + 10 + 2 + 4 = 20$. On the other hand, the cost of the plan $(T, T, S, T)$ is $50 + 20 + 20 + 10 + 4 = 114$. 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. Note that you will likely need two subproblems, which will be mutually recursive. [8 points]

4. You are a tourist visiting the planet Zondor and, since you are unfamiliar with Zondorian coins, you are burdened by way too many of them. The next time you buy something, you are determined to use as many of your coins as possible, and want to give the vendor exact change so you won’t receive any more coins as change.

For this problem, the Zondorian coins have integer values $z_1, z_2, \ldots, z_n$ and you can assume that you have a limitless number of each coin. When you purchase something of value $Y$, you want to determine the maximum number of coins you can give to purchase the item with exact change (you may return some dummy value, -1, 0, $\pm\infty$, if it is not possible to exactly match $Y$). For example, if $(z_1, z_2, z_3) = (3, 7, 11)$ and $Y = 25$, you could get rid of 3 coins (3, 3, 3, 3, 3, 3, 7).

So, the problem here is to perform the two steps above to start the process of solving the MAXZONDORCOIN problem. [8 points]

Total: 32 points