1. (10 points) Solve the all-pairs shortest-path problem for the digraph with the weight matrix shown below:

\[
D^{[0]} = \begin{pmatrix}
0 & 2 & \infty & 1 & 8 \\
6 & 0 & 3 & 2 & \infty \\
\infty & \infty & 0 & 4 & \infty \\
\infty & \infty & 2 & 0 & 3 \\
3 & \infty & \infty & \infty & 0
\end{pmatrix}
\]

(a) Draw the digraph given by this weight matrix, \(D^{[0]}\). Assume that the nodes are labeled A, B, C, D, E going across and down.

(b) Show all matrices \(D^{[1]}\) through \(D^{[5]}\). (\(D^{[0]}\) is given above.)

(c) Show all matrices \(\Pi^{[0]}\) through \(\Pi^{[5]}\).

(d) Write the path produced by following the Print-APSP algorithm, given in the lecture notes, between every pair of vertices in the graph.

NOTE: You have the option of writing Floyd’s algorithm in Java or Python and submitting the program and the results printed out of the program in a single folder. You can hard code the weight matrix and the predecessor matrix as instance variables in your program. If you choose to program this algorithm, printing the paths between each pair of vertices will be much simpler. See your professor if you want to compare results. You will receive more credit for the coded version.
2. Suppose you want to find the optimal way to parenthesize 4 matrices of the following sizes:

<table>
<thead>
<tr>
<th>matrix</th>
<th>$A_1$</th>
<th>$A_2$</th>
<th>$A_3$</th>
<th>$A_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension</td>
<td>$40 \times 20$</td>
<td>$20 \times 30$</td>
<td>$30 \times 10$</td>
<td>$10 \times 30$</td>
</tr>
</tbody>
</table>

Input: $p = (40, 20, 30, 10, 30)$

Using the Matrix-Chain-Order algorithm from our lecture notes and from pages 375–377 of our textbook, compute the optimal way to parenthesize the set of matrices such that the least number of multiplications are used. Fill in the upper-right diagonals of the $m$ and $s$ matrices given below.

Use the $s$ table as input to the Print-Optimal-Parens algorithm, with this call: Print-Optimal-Parens($s, 1, 4$). Use this algorithm to produce the optimal parenthesization of matrices $A_1$ through $A_4$.

Note: You have the option of writing the Matrix-Chain-Order algorithm in Java or Python and submitting the program and the results printed out of the program in a single folder. You can hard-code the dimension array as an instance variable in your program. See your professor if you want to compare results. You will receive more credit for the coded version.