1. Using the graph below, show the start and finish times determined by DFS. Using those finish times, provide a topological sort of the nodes of the graph. As on the homework, visit nodes in alphabetical order.
2. On hw3 we looked at a modification of Dijkstra’s algorithm to find reliable paths in a graph. The graph is $G = (V, E)$ with $V$ representing a set of locations and $E$ representing a communications channel between two points. We are also given a start point $s \in V$ and a reliability function $r : V \times V \rightarrow [0, 1]$. The goal is to find the reliability of the most reliable path from $s$ to all other nodes in $G$.

For any points $u, v \in V$, $r(u, v)$ is the probability that the communication link $(u, v)$ will not fail: $0 \leq r(u, v) \leq 1$. If there is a path with two edges, for example from $u$ to $v$ to $w$, then the reliability of that path is $r(u, v) \cdot r(v, w)$.

The modification was to use a MAX priority queue, initialize the reliability of all nodes to 0 ($\forall u \in V \ u.rel = 0$) and then set $s.rel = 1$. The RELAX method became

\[
\text{RELAX}(u,v)
\text{ if } v.rel < u.rel \cdot r(u,v)
\text{ then }
\quad v.rel = u.rel \cdot r(u,v)
\quad v.prev = u
\]

Here we will add a field $u.numPaths$ to all nodes $u \in V$. You need to modify RELAX to ensure that $u.numPaths$ will contain the number of distinct most reliable paths from $s$ to $u$. Be sure to also state how the $numPaths$ fields are initialized.
3. Illustrate the Floyd-Warshall algorithm on the graph with the following weight matrix

\[
\begin{pmatrix}
0 & 8 & \infty & \infty \\
8 & 0 & 5 & 3 \\
\infty & 5 & 0 & 6 \\
\infty & 3 & 6 & 0
\end{pmatrix}
\]

Show the intermediate matrices for \( k = 1, 2, 3, 4 \). Note that the graph is undirected (and hence the matrix is symmetric).
4. We need to place a sequence of billboards and are allowed to do so at certain specified locations along a road. Each location has a different cost, and we want to minimize the total cost of billboard placement. We cannot place none, since there is a penalty if the billboards are too far apart.

The billboards can be placed at mileposts $m_0, m_1, \ldots, m_n$ which have placement costs $c_0, c_1, \ldots, c_n$ respectively. We are required to place a billboard at locations $m_0$ and at $m_n$. The penalty we pay is $100 for every integer multiple of 20 miles: for example, if a billboard is placed 17 miles after the previous one, there is no penalty; if a billboard is placed 42 miles after the previous one, we pay a penalty of \( \lfloor \frac{42}{20} \rfloor \cdot 100 = 2 \cdot 100 = 200 \) dollars.

To start a dynamic programming solution, we define subproblem $BP(i)$ to be the minimum cost of billboard placement for billboards 0, 1, 2, \ldots, $i$, where a billboard is to be placed at locations $m_0$ and $m_i$.

**Give a recurrence relation for BP. Include the base case(s).**