Problem 1. Consider the weighted directed graph $G = (V, E)$ below.

Suppose that we run Bellman-Ford’s algorithm to find the shortest path distances from vertex $a$ to all the other vertices. Recall that the algorithm performs $|V| - 1$ rounds of edge relaxations, and maintains a $dist(v)$ value for every vertex $V$. Give all the $dist(v)$ values after each round of edge relaxations.

Problem 2. Consider the weighted directed graph $G = (V, E)$ below.

Assign ids 1, 2, 3, and 4 to vertices $a$, $b$, $c$, and $d$, respectively. Suppose that we run the Floyd-Warshall algorithm to find the shortest path distance between vertex $i$ and vertex $j$ for all $i, j \in [1, 4]$. Recall that the algorithm needs to compute $spdist(i, j \mid \leq k)$ for all $i, j, k \in [1, 4]$. Give the value of $spdist(i, j \mid \leq k)$ for each possible combination of $i, j, k$.

Problem 3. Recall that the rationale behind the Floyd-Warshall algorithm is the following recursive function:

$$spdist(i, j \mid \leq k) = \min \left\{ spdist(i, j \mid \leq k - 1), spdist(i, k \mid \leq k - 1) + spdist(k, j \mid \leq k - 1) \right\}$$

Give a proof of the above function’s correctness.

Problem 4. When we discussed Bellman-Ford’s algorithm in the lecture, we described how to compute the shortest path distances from the source vertex $s$ to the other vertices. Augment our description to produce also the shortest paths from $s$ to the other vertices. The final algorithm should still run in $O(|V||E|)$ time.