1. Suppose that the order in which resource requests are granted is predetermined. Give an example of a snapshot with a resource deadlock that is not discovered by the Bracha-Toueg deadlock detection algorithm.

Show that in case of a nondeterministic selection which resource request is granted, the deadlock in your example may be avoided. (12 pts)

2. Give a computation of Frederickson’s algorithm, on an undirected ring of size three and with $\ell = 2$, to show that a forward can be sent to a node that is not a child of the sender. (10 pts)

3. The well-known Dijkstra’s single-source shortest path algorithm for undirected weighted networks (for a uniprocessor setting) works as follows. Initially only the initiator is visited, and has distance 0. In subsequent rounds:

- each visited node $p$ computes for its unvisited neighbors $q$ as distance: the distance of $p$ plus the (positive) weight of the edge $pq$;
- if unvisited node $q$ is thus in this round awarded a minimal distance $m$ (among all unvisited nodes), then it is added to the visited nodes, with distance $m$ and parent $p$.

Develop a distributed version of Dijkstra’s algorithm. Also discuss the worst-case message and time complexity of your algorithm. (16 pts)
4. Apply the Itai-Rodeh election algorithm to the following anonymous directed ring, where initial random id’s have been chosen at the start of round 0. Give a computation where only two processes progress to round 1, in which they both choose an id $j < i$, and ultimately one process becomes the leader.

5. Consider a complete network of five processes. Apply the Chandra-Toueg 2-crash consensus algorithm, where initially four processes choose the value 0 and one process the value 1. Give a computation in which all correct processes decide for 1.

6. The logical clock values in the Ricart-Agrawala mutual exclusion algorithm are unbounded. Adapt the algorithm such that the range of these values becomes finite. \textit{(Hint: Use modulo arithmetic.)}

7. Let preemptive jobs $J_1$, $J_2$ and $J_3$ arrive at times 2, 1 and 0, respectively, with execution time 2. Let the priorities be $J_1 > J_2 > J_3$. Let $J_1$ and $J_3$ use resource $R$ for their entire execution. The jobs are executed using priority ceiling.

   How are the three jobs executed if the arrival of $J_1$ is known from the start? And how are they executed if the arrival of $J_1$ is not known before time 2? For the latter question, consider the cases with and without priority inheritance. \textit{(Explain your answers.)}