Problem 9.3 provides eight extra credit points. Thus, there are 32 points achievable on this assignment, but only 24 points are accounted for as 100%.

9.1. Problem (12)  
Dijkstra’s Token Termination Detection Algorithm II

Consider Dijkstra’s Token Termination Detection Algorithm II. All messages are guaranteed to be non–overtaking. That is, if a process sends two messages $M_1$ and $M_2$ to another process, $M_1$ is received before $M_2$.

a.) Prove the correctness of Dijkstra’s Token Termination Detection Algorithm II.

b.) Does it suffice in step (2) that a process $i$ is colored black only if it sends a message to a process $j$ with $j < i$?

c.) Does it suffice in step (2) that a process $i$ is colored black only if it sends a message?

9.2. Problem (12)  
More on Termination Detection

Here are three more procedures to detect termination. Are they correct? You may assume that messages are delivered in order.

a.) Process 1 inserts a token into the process ring. A process keeps the token as long as it has work to do and passes it along otherwise. Process 1 prepares the shutdown whenever receiving the token from process $p$.

b.) Process 1 inserts a white token into the process ring. A process colors the token black if it has work to do and passes it along. An idle process passes the token along without changing its color. Process 1 prepares the shutdown whenever receiving a white token from process $p$.

c.) Our last procedure works exactly as Dijkstra’s Token Termination Detection Algorithm I, however whenever a token is passed to an active process, this process colors the token black and passes it on immediately.
9.3. Problem (8)  

Early Termination Detection

a.) The weight procedure for termination detection in dynamic load balancing has to be modified to allow early termination when a process finds a solution. How should that be done?

b.) Does the implementation of Dijkstra’s algorithm have to be modified?