Algorithm 1 Solving mutual exclusion among $n$ processes $p_1, \ldots, p_n$: code for $p_i$

1. Shared variables:
2. atomic register $L_{Ai}$, initially 0 for each process $p_i$
3. atomic register $r_i$, initially $false$ for each process $p_i$

4. Code executed by $p_i$:  
5. Entry Section code  
6. write($r_i, true$)  
7. write($L_{Ai}, max(L_{A0}, \ldots L_{An-1}) + 1$)  
8. write($r_i, false$)  
9. for all $j = 1 \ldots n$ do  
10. await $r_j = false$  
11. await $L_{Aj} = 0 \lor (L_{Aj}, j) \geq (L_{Ai}, i)$  
12. Enter Critical section  
13. Exit section code  
14. $L_{Ai} = 0$

Does the above algorithm ensure mutual exclusion? If not, construct an execution where there exist two processes that are inside the critical section at the same time (50 points).