Concurrent Systems and Applications

A multi-threaded application is using a long linked list of integers. The list is accessed through synchronized methods on a ListSet object.

The list itself comprises a chain of ListNode objects in ascending numerical order. The chain always starts and ends with special sentinel nodes conceptually containing $-\infty$ and $+\infty$ respectively. This simplifies the implementation of operations on the list: they do not have to deal with inserting elements at the very start or at the very end.

(a) Sketch the definition of ListSet and ListNode as Java classes. You need only give appropriate field definitions and the implementation of an insert method on ListSet. [4 marks]

(b) An engineer suggests that, instead of holding a lock on a ListSet object, threads only need to lock a pair of ListNode objects in the region that they are working.

(i) Define methods lock and unlock for your ListNode class to allow a thread to acquire a mutual exclusion lock on a given node. [6 marks]

(ii) Show how your insert method could be updated to incorporate the engineer’s idea. [8 marks]

(iii) Do you think the new implementation will be faster than the original one? Justify your answer. [2 marks]