HW #4

Pt. 1:

The RIPR model states 4 conditions (reachability, infection, propagation, reveal) necessary for a failure to be observed. The first condition we observe is reachability and since it is stated that the test method for removeDuplicate() compiles and runs to completion we can say that the location of the fault has been reached. The problem with the test method have to do with the infection and propagation conditions. This means that the state of the the program must be incorrect and the state must cause some incorrect output. With the ‘add’ operation, strings are stored in the order that they are inserted which in this case [“orange”, “apple”, “apple”, “banana”]. As mentioned, the order of string should be maintained when the elements are withdrawn, so if it is assumed that removeDuplicates() works as it is described, the final state of the program should be [“orange”, “apple”, “banana”]. However this is not the case. The test method uses method assertTrue if the first word is “orange”. Since order is maintained when strings are added and withdrawn, the first element returned from getFirst() will remain orange, even if the removeDuplicate() works without error. Therefore, the final state of the program will be correct, making the current test method incorrect for testing removeDuplicate().