Vulnerabilities in the Queue Implementation

1. Can force Queue to have fewer elements than the net number of adds
   Work with three threads t1, t2, t3
   t1:
   add method:
   head == null => true
   t2:
   add method:
   head==null => true
   t1:
   add method continued:
   tail = n
   tail.value =newValue //queue has one element now
   t3:
   add method:
   head == null => false
   tail.next = n //queue has two elements now
   t2:
   add method continued:
   head=n
   tail = n
   tail.value =newValue //queue has only one element now (lost an element)

2. Can force Queue to have more elements than the net number of adds – can force add, then remove and Queue still contains an element
   Work with two threads t1, t2
   t1:
   add method:
   head == null => true
   t2:
   remove method
   head==null => true
   t1:
   add method continued:
   head=n; tail = n; tail.value =newValue
   t2:
   remove method continued
   return null
   //Queue has one element after operations add, remove

3. Can force a Queue operation to produce a NullPointerException
   Work with threads t1, t2, t3
   t1:
   add method:
      head==null => true
      head = n; tail = n; tail.value =new Value
   t2:
   add method:
      head == null => false
   t3:
   remove method:
      head == null => false
      Node n = head
      head = n.next
   //queue is now empty and head =tail = null
   t2:
   add method continued:
      head = n
      tail = n
      tail.value = newValue //NullPointerException