Node data contains (Physical_Time, Logical_Time, Message, Server ID)
Logical_time = 0

(Receiving trade from client)
      Create new node
      Copy data to new node (message, server ID)
      Set node time to logical_time
      Set physical_time to server physical time
      Logical_time += 1
      Print_Order

(Receiving trade from other server)
      Create new node
      Copy data to new node
      Set physical_time to server physical time
      if logical_time <= node_logical_time:
            logical_time = node_logical_time+1
      else:
            logical_time += 1
      Print_Order

(Print_Order)
      Remember head node
      if time_limit has passed on head:
            Organize Nodes by logical_time and server_id (in increasing order)
            Print all nodes that were received before or at the same time as when the head
            node was received and replace head

This algorithm will always print trades in the same order due to the use of logical clocks. A logical clock keeps track of time without using actual clocks by assigning each event a number, where events occuring after this number happen later. In our algorithm, whenever a trade is received we assign it a number from the logical clock and then increase the logical clock. This assigns an ordering for each event that occurs on the server. When receiving a trade from another server, the other server will have already ordered their trades. To retain the same order across servers, we keep their original logical times. However, it is possible for the other server's logical time to be ahead of our logical time. This means if we don't update our server's logical time when we receive a client trade, our ordering will incorrectly assume this trade happened before receiving the other server's trades and their orderings. In order to maintain the correct ordering of events for our server, we need to update our own server time to be greater than their logical time. This way if we receive events after receiving another server's ordering, it will not affect the overall ordering of our events. While we now have a way of ordering events, this means that our list will have different events with the same logical time. To differentiate and correctly order events for all servers, when printing these events in logical time order we print events with the same time in increasing server_id.